

TUGAS AKHIR - TE 141599

**DYNAMIC ECONOMIC DISPATCH DENGAN
MEMPERTIMBANGKAN VALVE POINT EFFECT DAN MULTIPLE
FUEL OPTIONS MENGGUNAKAN METODA ANT COLONY
OPTIMIZATION**

Mochamad Luthfiyansah
NRP 2215 105 063

Dosen Pembimbing
Dr. Rony Seto Wibowo, ST. MT.
Prof. Dr. Ir. Adi Soeprijanto, MT.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR - TE 141599

***DYNAMIC ECONOMIC DISPATCH DENGAN
MEMPERTIMBANGKAN VALVE POINT EFFECT DAN
MULTIPLE FUEL OPTIONS MENGGUNAKAN METODA
ANT COLONY OPTIMIZATION***

Mochamad Luthfiyansah
NRP 2215105063

Dosen Pembimbing
Dr. Rony Seto Wibowo, ST. MT.
Prof. Dr. Ir. Adi Soeprijanto, MT.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2017



FINAL PROJECT - TE 141599

***DYNAMIC ECONOMIC DISPATCH CONSIDER VALVE
POINT EFFECT AND MULTIPLE FUEL OPTIONS USING
ANT COLONY OPTIMIZATION METHOD***

Mochamad Luthfiyansah
ID 2215105063

Advisor
Dr. Rony Seto Wibowo, ST. MT.
Prof. Dr. Ir. Adi Soeprijanto, MT.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2017

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "*Dynamic Economic Dispatch Dengan Mempertimbangkan Valve Point Effect Dan Multiple Fuel Options Menggunakan Metode Ant Colony Optimization*" adalah benar benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juni 2017



Mochamad Luthfiyansah
2215105063

Halaman ini sengaja dikosongkan

**DYNAMIC ECONOMIC DISPATCH DENGAN
MEMPERTIMBANGKAN VALVE POINT EFFECT DAN
MULTIPLE FUEL OPTIONS MENGGUNAKAN METODA
ANT COLONY OPTIMIZATION**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik**

Pada

**Bidang Studi Teknik Sistem Tenaga
Departemen Teknik Elektro
Institut Teknologi Sepuluh Nopember**

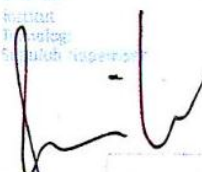
Menyetujui :

Dosen Pembimbing I



Dr. Rony Seto Wibowo, ST., MT.
NIP. 19741129 200012 1 001

Dosen Pembimbing II



Prof. Dr. Ir. Adi Soeprijanto, MT.
NIP. 19640405 199002 1 001



JULI, 2017

Halaman ini sengaja dikosongkan

***DYNAMIC ECONOMIC DISPATCH DENGAN
MEMPERTIMBANGKAN VALVE POINT EFFECT
DAN MULTIPLE FUEL OPTIONS
MENGUNAKAN METODA ANT COLONY
OPTIMIZATION***

Nama : M. Luthfiyansah
NRP : 2215105063
Dosen Pembimbing 1 : Dr. Rony Seto Wibowo, ST. MT.
Dosen Pembimbing 2 : Prof. Dr. Ir. Adi Soeprijanto, MT.

ABSTRAK

Biaya pembangkitan yang ekonomis pada pembangkit thermal sulit untuk ditentukan karena banyak faktor yang perlu dipertimbangkan. Economic dispatch adalah solusi untuk mendapatkan biaya pembangkitan yang ekonomis dan efisien yaitu dengan perhitungan optimisasi biaya. Economic dispatch hanya digunakan untuk satu level beban. Dynamic Economic Dispatch (DED) merupakan solusi untuk menentukan pembagian pembebanan dari setiap unit pembangkit secara ekonomis dalam rentang waktu 24 jam. Beberapa faktor yang perlu dipertimbangkan dalam perhitungan DED yaitu valve point effect (VPE) merupakan efek dari pembukaan katub bahan bakar pada unit pembangkit thermal dan multiple fuel options (MFO). parameter tersebut menjadikan kurva biaya pembangkitan menjadi non-convex. Dengan menggunakan metode optimisasi didapatkan biaya seminimal mungkin serta tetap melayani kebutuhan daya ke konsumen selama 24 jam. Pada tugas akhir ini metode optimisasi yang digunakan yaitu Ant Colony Optimization. Fungsi objektif yang dioptimisasi adalah biaya pembangkitan minimal dengan pertimbangan VPE dan MFO. Hasil pengujian simulasi yang didapatkan pada sistem 5 Unit pembangkit dan 10 Unit pembangkit yaitu pengaruh dari pertimbangan VPE dan MFO pada fungsi objektif yang disediakan. Sehingga menghasilkan perubahan biaya pembangkitan yang signifikan untuk memenuhi kebutuhan konsumen selama 24 jam.

Kata Kunci : *Dynamic Economic Dispatch, Valve Point Effect, Multiple Fuel Options, Ant Colony Optimization*

Halaman ini sengaja dikosongkan

DYNAMIC ECONOMIC DISPATCH CONSIDER VALVE POINT EFFECT AND MULTIPLE FUEL OPTIONS USING ANT COLONY OPTIMIZATION METHOD

Name : M. Luthfiyansah
ID : 2215105063
Supervisor 1 : Dr. Rony Seto Wibowo, ST. MT.
Supervisor 2 : Prof. Dr. Ir. Adi Soeprijanto, MT.

ABSTRACT

The economical generation cost of a thermal generator is difficult to determine due to many factors to consider. Economic dispatch is a solution to get economical and efficient cost of generation that is with the calculation of cost optimization. Economic dispatch is only used for one load level. Dynamic Economic Dispatch (DED) is a solution to determine the distribution of the load of each generating unit economically within 24 hours. Some factors to consider in DED calculations are the valve point effect (VPE) is the effect of opening the fuel valve on thermal generating units and multiple fuel options (MFO). These parameters make the generation cost curve to be non-convex. By using the optimization method, it can get the minimum cost and keep serving the power requirement to the consumer for 24 hours. In this final project, the optimization method used is Ant Colony Optimization. Optimized objective function is minimum generation cost considering VPE and MFO. Result of simulation test which got in 5 generator unit and 10 generating unit that is influence from consideration of VPE and MFO on objective function provided. This results in significant changes in generating costs to meet customer needs over the next 24 hours.

***Keywords : Dynamic Economic Dispatch, Valve Point Effect,
Multiple Fuel Options, Ant Colony Optimization***

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Penulis dengan segala kerendahan hati melantunkan puji syukur Alhamdulillah atas kehadiran Allah SWT yang telah mencurahkan hidayah dan karunia-Nya, sehingga penyusunan laporan tugas akhir yang berjudul ***“Dynamic Economic Dispatch Dengan Mempertimbangkan Valve Point Effect Dan Multiple Fuel Options Menggunakan Metode Ant Colony Optimization”*** dapat diselesaikan dengan baik. Laporan ini disusun sebagai salah satu persyaratan untuk menyelesaikan jenjang pendidikan S1 pada Bidang Studi Teknik Sistem Tenaga, Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember, Surabaya. Atas selesainya penyusunan tugas akhir ini, penulis ingin mengucapkan terima kasih kepada:

1. Allah SWT atas segala rahmat, rejeki dan hidayahnya yang diberikan sehingga laporan tugas akhir dapat terselesaikan.
2. Ayah, Ibu dan kedua kakak tercinta yang memberikan doa dan memberikan dukungan serta motivasi kepada penulis baik dalam keadaan senang ataupun susah.
3. Dr. Rony Seto Wibowo, ST., MT. dan Prof. Dr. Ir. Adi Soeprijanto, MT selaku dosen pembimbing yang telah memberikan arahan, bimbingan dan perhatiannya selama proses penyelesaian tugas akhir ini.
4. Teman-teman LBE Dr. Rony Seto Wibowo, ST., MT. yang saling memberikan support dan menjadi tempat bertukar pendapat selama pengerjaan tugas akhir ini.
5. Teman-teman seangkatan yang menjadi partner diskusi dan belajar selama 2 (dua) tahun masa studi penulis serta memberi semangat kepada penulis dalam penyelesaian tugas akhir ini.

Semoga tugas akhir ini bermanfaat bagi mahasiswa maupun peneliti. Penulis menyadari bahwa masih banyak kekurangan, untuk itu kritik dan saran diharapkan untuk penyempurnaan tugas akhir ini.

Surabaya, Juni 2017

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN.....	iii
HALAMAN PENGESAHAN.....	v
ABSTRAK	vii
ABSTRACT.....	ix
KATA PENGANTAR	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xix
 BAB 1 PENDAHULUAN	 1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Metode Penelitian.....	3
1.6 Sistematika Laporan	4
1.7 Relevansi atau Manfaat.....	5
 BAB 2 DYNAMIC ECONOMIC DISPATCH	
MEMPERTIMBANGKAN VALVEPOINT EFFECT	
DAN MULTIPLE FUEL OPTIONS.....	7
2.1 Sistem Tenaga Listrik	7
2.1.1 Generator	8
2.1.2 Sistem Transmisi	8
2.1.3 Sistem Distribusi	8
2.1.4 Beban	9
2.2 Karakteristik Pembangkit Listrik	9
2.3 Unit Pembangkit Thermal.....	10
2.3.1 Karakteristik Input-Output Pembangkit Thermal	11
2.3.2 Karakteristik Kenaikan Biaya/Panas Unit Thermal	12
2.4 Economic Dispatch.....	13
2.4.1 Valve Point Effect	14
2.4.2 Multiple Fuel Options.....	15
2.5 Dynamic Economic Dispatch.....	16
2.6 Ant Colony Optimization.....	17
2.6.1 Graf ACO.....	18

2.6.2 State Transition Rule [9,10].....	18
2.6.3 Local Pheromone Update [9].....	19
2.6.4 Global Pheromone Update [9]	19
BAB 3 IMPLEMENTASI ANT COLONY OPTIMIZATION	
PADA PERMASALAHAN DED.....	21
3.1 Penerapan Algoritma Ant Colony Optimization.....	21
3.2 Inisialisasi Permasalahan DED	23
3.2.1 Fungsi Objektif	23
3.2.2 Konstrai Kesetimbangan Daya.....	23
3.2.3 Konstrai Batasan Oprasional.....	23
3.3 Perancangan Plant	24
3.3.1 Sistem 5 Unit Pembangkit	24
3.3.2 Sistem 10 Unit Pembangkit	25
3.4 Algoritma ACO untuk Permasalahan DED	27
3.5 Parameter Ant Colony Optimization	27
3.6 Kontruksi Solusi yang Memungkinkan	31
3.6.1 Pemilihan Solusi Terbaik.....	32
3.7 Reduksi Variabel.....	33
3.8 Aturan Mini-Maxi Ant System	33
3.9 Memori Penyimpanan Semut.....	34
BAB 4 SIMULASI DAN ANALISA	35
4.1 Validasi Program ACO.....	35
4.2 Sistem Tenaga Listrik 5 Unit Pembangkit IEEE	37
4.2.1 Simulasi Kasus I Plant 1	37
4.2.2 Simulasi Kasus II Plant 1.....	41
4.3 Sistem Tenaga Listrik 10 Unit Pembangkit IEEE.....	45
4.3.1 Simulasi Kasus III Plant 2	45
4.3.1 Simulasi Kasus IV Plant 2	50
BAB 5 PENUTUP.....	57
5.1. Kesimpulan.....	57
5.2. Saran	58
DAFTAR PUSTAKA	59
LAMPIRAN.....	61
I. Tampilan GUI input Matlab	61
II. Tampilan GUI output Matlab	62

III.	M-file Ant Colony Optimization	63
IV.	M-file GUI input Matlab.....	72
V.	M-file GUI output Matlab.....	79
RIWAYAT PENULIS		85

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 2. 1 Sistem Tenaga Listrik.....	7
Gambar 2. 2 Sistem Unit Pembangkit Thermal	10
Gambar 2. 3 Kurva Karakteristik Input-Output Unit Thermal	11
Gambar 2. 4 Kurva Karakteristik Kenaikan Biaya	12
Gambar 2. 5 Kurva Biaya Bahan Bakar dengan 3 Katup	15
Gambar 2. 6 Kurva Biaya dengan Suplai 3 Type Bahan Bakar.....	16
Gambar 3. 1 Flowchart Algoritma ACO	22
Gambar 3. 2 Flowchart Implementasi ACO Pada DED.....	28
Gambar 3. 3 Flowchart Implementasi ACO Pada DED Lanjutan ...	29
Gambar 3. 4 Graf ACO.....	30
Gambar 4. 1 Kurva Permintaan Beban 24 Jam	37
Gambar 4. 2 Kurva Pembangkitan pada Generator 2.....	39
Gambar 4. 3 Kurva Konvergensi IEEE 5 Unit Generator.	40
Gambar 4. 4 Kurva Kestimbangan Daya Pembangkitan dan Pembebanan.....	40
Gambar 4. 5 Kurva Pembangkitan pada Generator 5 dengan 3 Katup Pembukaan	43
Gambar 4. 6 Konvergensi IEEE 5 Unit Generator dengan Valve Point Effect.	44
Gambar 4. 7 Kurva Kestimbangan Daya Pembangkitan dan Pembebanan.....	44
Gambar 4. 8 Kurva Permintaan Beban Satu Periode	47
Gambar 4. 9 Kurva Pembangkitan pada Generator 2 dengan 4 Katup Pembukaan	48
Gambar 4. 10 Konvergensi IEEE 10 Unit Generator dengan Valve Point Effect.	49
Gambar 4. 11 Kurva Kestimbangan Daya Pembangkitan dan Pembebanan.....	49
Gambar 4. 12 Kurva Pembangkitan pada Generator 2 dengan pertimbangan VPE dan MFO	52
Gambar 4. 13 Konvergensi IEEE 10 Unit Generator dengan VPE dan MFO	53
Gambar 4. 14 Kurva Kestimbangan Daya Pembangkitan dan Pembebanan.....	54
Gambar 4. 15 Perbandingan Parameter Kurva Pembangkitan pada Generator 2	55

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 3. 1 Data Fungsi Biaya Sistem 5 Unit Pembangkit	24
Tabel 3. 2 Data Generator Sistem 5 Unit Pembangkit	24
Tabel 3. 3 Data Generator Sistem 5 Unit Pembangkit	25
Tabel 3. 4 Data Fungsi Biaya Sistem 10 Unit Pembangkit	26
Tabel 3. 5 Data Generator Sistem 10 Unit Pembangkit	26
Tabel 3. 6 Data Beban Sistem 10 unit pembangkit	26
Tabel 3. 6 Data Beban Sistem 10 unit Pembangkit (lanjutan)	27
Tabel 3. 7 Tabel Korelasi Parameter ACO	27
Tabel 4. 1 Data Pembangkit Example 3A.....	35
Tabel 4. 2 Data Fungsi Biaya Example 3A.....	36
Tabel 4. 3 Hasil Simulasi DED 5 Unit Pembangkit	38
Tabel 4. 4 Komparasi Metode untuk 5 Unit Pembangkit	41
Tabel 4. 5 Hasil Simulasi DED 5 Unit Pembangkit dengan VPE....	41
Tabel 4. 6 Tabel Perbandingan Pada Plant I	43
Tabel 4. 7 Hasil Simulasi DED (24 jam) Kasus III Plant 2 Bagian 1	45
Tabel 4. 8 Hasil Simulasi DED (24 jam) Kasus IV Plant 2 Bagian 150	
Tabel 4. 9 Perbandingan Pada Plant II	53
Tabel 4. 10 Komparasi Metode untuk 10 Unit Pembangkit	54

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi yang diiring dengan bertambahnya jumlah penduduk tentu menyebabkan permintaan daya listrik yang terus bertambah. Peran pembangkit untuk mensuplai permintaan daya listrik yang terus bertambah diupayakan mampu bekerja secara maksimal dan efisien dengan biaya pembangkitan yang minimum untuk memenuhi permintaan beban. Namun meningkatnya permintaan konsumen yang terus meningkat untuk pemakaian listrik menjadi kendala pada sisi pembangkitan, ditambah akibat dari fluktuasi kebutuhan daya listrik di sisi beban menyebabkan suplai tenaga listrik menjadi kebutuhan utama [1]. Untuk mencapai kebutuhan konsumen saat ini pengoptimalan generator dalam membangkitkan daya dengan harga minimum dalam sistem tenaga listrik sangat diperlukan, karena proses pembangkitan dan penyaluran dalam sistem tenaga listrik untuk permintaan konsumen memerlukan biaya yang sangat besar. Unit pembangkit dengan beberapa pilihan bahan bakar adalah salah satu cara untuk mengurangi biaya pembangkit listrik. Pada pembangkit listrik tersebut, biaya pembangkitan minimum dicapai dengan penentuan jenis bahan bakar yang optimal [1,2]. Koordinasi antar pembangkitan diperlukan dalam upaya melakukan optimal generator *scheduling* untuk memperoleh biaya yang minimum.

Dynamic Economic Dispatch (DED) merupakan suatu hal penting dalam operasi pada sistem tenaga listrik. Fungsi utama DED sebagai perhitungan untuk membangkitkan daya dari setiap pembangkit yang beroperasi untuk dapat memenuhi kebutuhan beban pada biaya pembangkitan paling minimal secara real time pada sistem tenaga. *Economic Dispatch* (ED) konvensional hanya dapat digunakan untuk satu level beban. Jika digunakan pada sistem dengan beban dalam rentang waktu tertentu akan ada beberapa generator yang menyalahi parameter *ramp rate* dari generator [2]. Dengan parameter *ramp rate*, ED tidak dapat diselesaikan pada satu level beban. DED merupakan pengembangan dari ED konvensional karena memperhitungkan batasan *ramp rate* dari unit pembangkit DED digunakan untuk menentukan pembagian pembebanan unit pembangkit secara ekonomis dalam rentang waktu tertentu tanpa melanggar batasan *ramp rate* dari unit pembangkit. Parameter *ramp rate* menyebabkan pembagian pembebanan pada satu waktu tertentu

akan mempengaruhi pembagian pembebanan pada waktu lain [2,3]. Parameter lain yang akan dipertimbangkan adalah efek dari katup bahan bakar untuk memutar turbin serta beberapa pilihan bahan bakar yang optimal.

Fungsi biaya pembangkit diperoleh dari titik data yang diambil selama "*heat run*" tes, yaitu ketika input dan output data diukur sebagai suatu unit yang secara perlahan bervariasi melalui wilayah operasinya. *Wire drawing effect* yang terjadi karena masing-masing katup uap masuk dalam turbin mulai membuka, menghasilkan efek *ripple* pada kurva unit pembangkit [4]. Untuk mempertimbangkan kurva biaya yang akurat dari setiap unit pembangkit, efek katup-titik (*Valve Point Effect*) harus dimasukkan dalam model fungsi biaya. Oleh karena itu, fungsi sinusoidal dimasukkan ke dalam fungsi kuadrat [6].

Selain itu, terdapat unit pembangkit dalam suatu area yang dipasang dengan sumber multi-bahan bakar (batubara, gas alam, atau minyak), menyebabkan masalah penentuan bahan bakar paling ekonomi untuk proses pembakaran. Unit pembangkit dengan beberapa pilihan bahan bakar menghasilkan kurva biaya yang tidak mulus. Fungsi biaya yang dihasilkan disebut "fungsi biaya *hybrid*." Setiap segmen dari fungsi biaya hybrid terdiri dari beberapa informasi tentang bahan bakar yang dibakar atau operasi dari tiap-tiap unit. Karena unit pembangkit yang disediakan dengan sumber multi-bahan bakar, masing-masing unit harus diwakili dengan beberapa fungsi kuadrat *piecewise* untuk mencerminkan efek dari perubahan jenis bahan bakar, dan generator harus mengidentifikasi bahan bakar yang paling ekonomi untuk membakar dengan perhitungan yang akurat [4,5].

Pada tugas akhir ini digunakan salah satu metode *Ant Colony Optimization* (ACO) yang kemudian dilakukan perbandingan tanpa memperhatikan parameter *valve point effect* (VPE) dan *multiple fuel options* (MFO) serta dengan memperhatikan parameter VPE dan MFO untuk mendapatkan biaya pembangkitan yang minimal.

1.2 Permasalahan

Permasalahan yang akan dibahas dalam tugas akhir ini adalah menyelesaikan perhitungan DED dengan algoritma ACO dengan beberapa pertimbangan.

1. Menentukan pengaruh biaya dan pembebanan unit generator tanpa mempertimbangkan VPE dan MFO.
2. Menentukan pengaruh biaya dan pembebanan unit generator jika dengan mempertimbangkan VPE dan MFO.

1.3 Batasan Masalah

Dalam pengerjaan tugas akhir ini, permasalahan dibatasi sebagai berikut:

1. Tidak membahas detail tentang kestabilan tegangan.
2. Semua pembangkit diasumsikan selalu dalam keadaan menyala.
3. Data beban tidak boleh melebihi kemampuan pembangkitan total pembangkit.
4. Pengimplementasian menggunakan metode *Ant Colony Optimization* pada perhitungan DED dilakukan dengan MATLAB.

1.4 Tujuan Penelitian

Tujuan dari tugas akhir ini

1. Melakukan optimisasi pada pembagian beban unit pembangkit secara ekonomis dan efisien
2. Mendapatkan biaya pembangkitan paling ekonomis melalui perhitungan DED menggunakan metode *Ant Colony Optimization* dengan memperhatikan batasan dan permasalahan yang ada.

1.5 Metode Penelitian

Penelitian ini dilakukan melalui beberapa tahapan metodologi yaitu studi literatur, pengadaan journal ilmiah, pengadaan proceeding, pengadaan text book, dan metode optimasi.

1. Studi Literatur

Pada tahap ini akan dipelajari mengenai karakteristik pembangkit, teori penunjang studi analisa sistem tenaga dan metode program *Ant Colony Optimization*.

2. Identifikasi Sistem

Pada ini dilakukan deskripsi mengenai karakteristik pembangkit, fungsi biaya pembangkit, daya maksimal dan minimal serta *ramp-rate* pembangkit.

3. Pembuatan Program

Pada tahap pembuatan program menggunakan metode *Ant Colony Optimization* untuk menyelesaikan permasalahan *Dynamic Economic Dispatch* dilakukan dalam M-File pada MATLAB serta dibuat *Graphical User Interface (GUI)* untuk mempermudah pengoprasian program.

4. Simulasi dan Analisi

Melakukan simulasi *dynamic economic dispatch* menggunakan metode *ant colony optimization* dengan data

yang digunakan dalam penelitian ini diambil dari paper M. Q. Wang, X. S. Han, M. Yang and M. X. Wang, “*Dynamic Economic Dispatch with Valve Point Effect, Prohibited Operation Zones, and Multiple Fuel Option*,” *IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*., pp. 1-5, Feb. 2014 dan juga data *plant* yang akan digunakan sebagai uji coba.

5. Penulisan Buku

Penulisan laporan dilakukan sebagai penggambaran kesimpulan dari tugas akhir ini. Kesimpulan tersebut merupakan jawaban dari permasalahan yang dianalisis. Selain itu juga akan diberikan saran sebagai masukan berkaitan dengan apa yang telah dilakukan

1.6 Sistematika Laporan

Pembahasan tugas akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini membahas tentang dasar penyusunan Tugas Akhir yang meliputi latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, sistematika penyusunan laporan Tugas Akhir, dan relevansi.

Bab II Teori Dasar

Bab ini berisi tentang dasar teori yang merupakan penjelasan mengenai sistem kelistrikan secara umum, economic dispatch, dynamic economic dispatch dan metode *Ant Colony Optimization* untuk penerapan economic dispatch.

Bab III Perancangan Sistem

Bab ini berisi tentang uraian tentang perancangan, pembuatan dan penerapan *Ant Colony Optimization* pada permasalahan *Dynamic Economic Dispatch* dengan menggunakan MATLAB.

Bab IV Pengujian Sistem

Bab ini berisi tentang hasil pengujian *software* terhadap *Dynamic Economic Dispatch* dengan pertimbangan *Valve Point Effect* dan *Multiple Fuel Options*.

Bab V Penutup

Bab ini berisi kesimpulan dan saran-daran dari tugas akhir yang telah dikerjakan.

1.7 Relevansi atau Manfaat

Hasil yang diperoleh dari Tugas Akhir ini diharapkan menjadi sebuah gambaran mengenai penerapan metode *Ant Colony Optimization* dalam menghitung *Dynamic Economic Dispatch* dengan mudah. Selain itu, diharapkan menjadi salah satu cara dalam menentukan penjadwalan pembangkit yang optimal dengan Batasan-batasan yang digunakan.

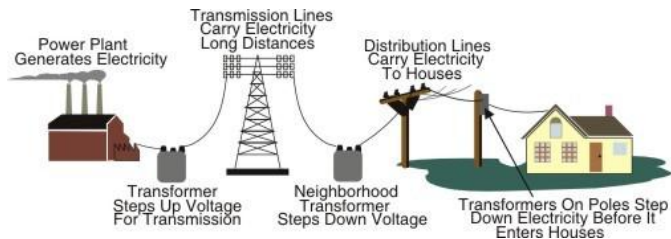
Halaman ini sengaja dikosongkan

BAB 2

DYNAMIC ECONOMIC DISPATCH MEMPERTIMBANGKAN VALVE POINT EFFECT DAN MULTIPLE FUEL OPTIONS

2.1 Sistem Tenaga Listrik

Sistem tenaga listrik merupakan suatu sistem penyediaan tenaga listrik mulai dari produsen listrik sampai ke konsumen listrik. Secara umum sistem tenaga listrik dibagi menjadi tiga bagian diantaranya pembangkit tenaga listrik, penyaluran tenaga listrik, dan distribusi tenaga listrik. Ketiga bagian tersebut sangatlah penting dalam penyediaan tenaga listrik. Karena merupakan sebuah kesatuan dari sistem kompleks yang bekerja untuk menyalurkan energi listrik dari pusat pembangkit ke beban-beban konsumen.



Gambar 2. 1 Sistem Tenaga Listrik

Gambar 2.1 menunjukkan sebuah sistem tenaga listrik secara sederhana mulai dari energi listrik yang dihasilkan pusat pembangkit listrik selanjutnya disalurkan melalui saluran transmisi kemudian melalui saluran distribusi akan sampai ke konsumen.

Pusat pembangkit listrik (*Power Plant*) merupakan tempat dibangkitkannya energi listrik dari turbin yang bergerak memutar untuk memutar generator sehingga menghasilkan energi listrik. Terdapat jenis pusat pembangkit listrik yang terbagi menjadi dua bagian besar yaitu pembangkit *hydro* (PLTA) dan pembangkit *thermal* (PLTU, PLTD, PLTG dan PLTP) [1].

Transmisi tenaga listrik merupakan proses penyaluran tenaga listrik dari pusat pembangkit hingga saluran distribusi dengan jarak relatif jauh. Sebelum masuk bagian transmisi tenaga listrik tegangan dinaikkan terlebih dahulu dengan trafo *step up* untuk mengatasi *losses*. Standarisasi *range* tegangan transmisi di Indonesia yaitu 500 kV untuk Saluran Tegangan Tinggi [2].

2.1.1 Generator

Salah satu bagian besar dari sistem tenaga listrik adalah stasiun pembangkit tenaga listrik. Stasiun pembangkit tenaga listrik tersebut dapat berupa generator yang digerakkan dengan tenaga gas, tenaga air, tenaga diesel dan lain sebagainya.

Generator dalam membangkitkan energi listrik melalui berbagai macam proses pembangkitan tenaga listrik. Energi dari bahan bakar yang digunakan oleh generator diubah oleh *primeover* menjadi energi mekanis, atau berasal dari energi mekanik turbin. Sehingga putaran dari turbin yang kemudian dapat menghasilkan energi listrik menggunakan induksi elektromekanik. Proses ini dikenal sebagai pembangkitan tenaga listrik [1,2].

2.1.2 Sistem Transmisi

Sistem transmisi berperan sebagai penyalur tenaga listrik yang telah dibangkitkan oleh generator sebagai pembangkit untuk disalurkan ke pusat beban melalui saluran transmisi. Sebelum disalurkan ke pusat beban, dalam sistem transmisi daya listrik di transformasikan dari tegangan tinggi ke tegangan rendah atau sebaliknya melalui gardu induk. Saluran transmisi juga menghubungkan peralatan sistem kerja normal maupun ketika terjadi gangguan.

Standart tegangan transmisi yang dikeluarkan oleh ANSI dioperasikan lebih dari 60 kV. Dengan standart satuan 69 kV, 115 kV, 138 kV, 161 kV, 230 kV, 345 kV, 500 kV, and 765 kV line-to-line. Tegangan transmisi diatas 230 kV disebut sebagai tegangan ekstra tinggi (Extra High Voltage-EHV). Alasan menggunakan tegangan ekstra tinggi untuk mengurangi rugi-rugi yang disebabkan oleh arus yang mengalir pada kabel penghantar pada sistem.

2.1.3 Sistem Distribusi

Sistem distribusi adalah bagian penting dari sub sistem tenaga listrik yang langsung berhubungan dengan pelanggan/konsumen. Pada sistem distribusi umumnya yaitu dengan merubah tegangan ekstra tinggi dari sistem transmisi tegangan tinggi menjadi tegangan menengah yaitu kurang dari atau sama dengan tegangan 20 kV. Fungsi utama sistem distribusi adalah untuk pembagian atau penyaluran tenaga listrik ke beberapa tempat. Sub sistem ini terdiri dari pusat pengatur beban / gardu induk, saluran tegangan menengah (6 kV dan 20 kV), jaringan sekunder (380 V dan 220 V) [3].

2.1.4 Beban

Suatu faktor utama yang paling penting dalam perencanaan sistem distribusi adalah karakteristik dari berbagai beban. Penentuan karakteristik beban listrik pada gardu distribusi bertujuan untuk meng evaluasi pembebanan gardu distribusi ataupun dalam merencanakan untuk penambahan gardu distribusi yang baru.

Pada dasarnya setiap sektor beban memiliki karakteristik yang berbeda – beda, mengingat berkaitan dengan pola konsumsi energi pada masing masing konsumen disetiap sekotr tersebut, seperti sektor perumahan, sektor industri, sektor komersial, dan sektor usaha. Sehingga dalam waktu 24 jam terbagi menjadi dua, yaitu beban dasar pada siang hari dan beban puncak pada sore hingga malam hari.

2.2 Karakteristik Pembangkit Listrik

Karakteristik pembangkit listrik mempermudah untuk penekanan dalam menentukan pembiayaan bahan baku energi. Output pembangkit dapat diatur dengan baik sehingga biaya bahan baku energi dapat diminimalisir yaitu dengan mengenal karakteristik dari pembangkit listrik. Permodelan secara matematis dari karakteristik pembangkit listrik bertujuan untuk mempermudah proses optimasi aga dihasilkan biaya pembangkitan yangh ekonomis dan optimal.

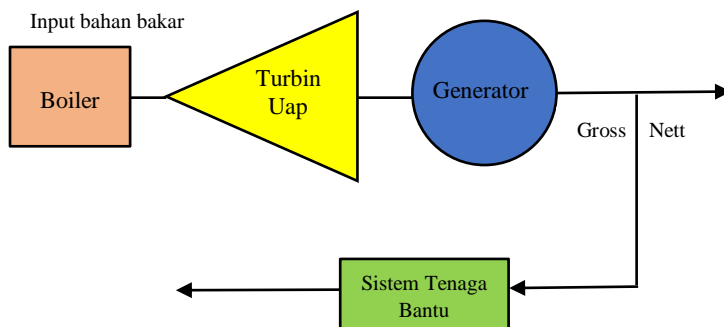
Terdapat berbagai macam jenis pembangkit listrik yang di klasifikasikan menjadi dua bagian besar berdasarkan karakteristiknya yaitu pembangkit *hydro* dan pembangkit *thermal*. Namun, di antara dua karakteristik pembangkit tersebut, pembangkit *thermal* cukup menjadi perhatian dikarenakan biaya bahan bakarnya yang tinggi dan cukup sering berubah untuk memenuhi kebutuhan beban. Oleh karena itu, untuk lebih lanjutnya akan banyak dibahas mengenai pembangkit *thermal* ini [4].

Secara umum jenis-jenis posisi pembangkit dalam sistem adalah pembangkit pemikul beban dasar (*base load*), pemikul beban menengah (*load follower*) dan pemikul beban puncak (*peaker*). Pembangkit yang digolongkan ke dalam pembangkit *base load* adalah pembangkit yang dapat beroperasi *on* atau *off* dalam waktu singkat dengan biaya oprasi yang murah. Sehingga pembangkit *base load* dioperasikan pada kapasitas maksimum sepanjang pembangkit tersebut siap dalam penyaluran daya, seperti PLTU batubara. Pembangkit jenis selanjutnya yaitu *load follower* meliputi pembangkit yang lebih fleksibel namun lebih mahal dari pembangkit *base load*, seperti PLTGU gas dan PLTU minyak. Pembangkit yang difungsikan sebagai pemikul beban puncak atau jenis pembangkit *peaker* meliputi pembangkit yang fleksibel baik dalam kecepatan perubahan

pembebanan maupun beroperasi *on* dan *off*. Pembangkit jenis ini seperti PLTG (gas), PLTD (diesel) [2]. Pembangkit dengan karakteristik yang kurang fleksibel yaitu pembangkit yang tidak dapat *on* atau *off* dalam waktu singkat dan lambat dalam menaikkan atau menurunkan pembebanan. Sehingga mengharuskan pembangkit untuk dioperasikan sepanjang pembangkit waktu.

2.3 Unit Pembangkit Thermal

Pembangkit termal adalah jenis pembangkit listrik yang pada umumnya menggunakan bahan bakar sumber energi panas dengan menggunakan boiler untuk proses pemanasan untuk menghasilkan uap panas bertekanan yang nantinya digunakan untuk menggerakkan turbin. Skema pembangkitan thermal digambarkan pada Gambar 2.2 terdiri dari boiler, turbin dan generator. Operasi ini merupakan pembangkit yang umum masih beroperasi saat ini, dimana PLTU membutuhkan bahan bakar fosil baik batubara maupun minyak bumi untuk menghasilkan listrik. Penentuan biaya pembangkitan dari unit pembangkit *thermal* berdasarkan karakteristik unit *input-output* dari pembangkit tersebut. Karakteristik *input-output* unit pembangkit termal dapat digambarkan melalui perubahan kurva laju panas yang direpresentasikan sebagai input total yang diukur dalam rupiah per jam dan output daya listrik dalam MW yang disediakan oleh sistem pembangkit tenaga listrik.



Gambar 2. 2 Sistem Unit Pembangkit Thermal

2.3.1 Karakteristik Input-Output Pembangkit Thermal

Karakteristik Input-Output Pembangkit *Thermal* merupakan dasar penyusunan dari fungsi biaya. Persamaan karakteristik input-output pembangkit menyatakan hubungan antara jumlah bahan bakar yang dibutuhkan untuk menghasilkan daya tertentu pada pembangkit listrik yang didekati dengan fungsi binomial, yaitu:

$$F_i(P_i) = a_i P_i^2 + b_i P_i + c_i \quad (2.1)$$

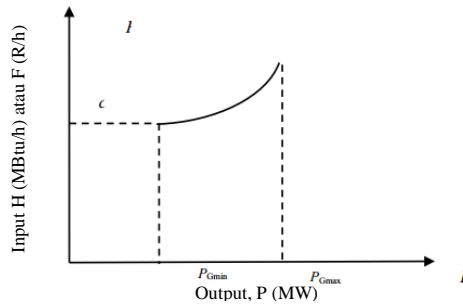
Keterangan:

F_i = Biaya bahan bakar perjam ke- i (Rp/jam)

P_i = Output daya pembangkitan ke- i (MW)

a_i, b_i, c_i = konstanta persamaan

Selain biaya bahan bakar yang dikonsumsi, biaya operasi juga meliputi biaya tenaga kerja, biaya pemeliharaan, biaya transportasi bahan bakar, dan lain lain. Biaya-biaya tersebut sulit direpresentasikan secara langsung sebagai fungsi biaya dari daya output yang dihasilkan generator. Karena permasalahan tersebut, biaya-biaya tersebut diasumsikan sebagai bagian fixed cost dari biaya operasi, dan akan diabaikan dalam proses optimisasi pada Tugas Akhir ini.



Gambar 2. 3 Kurva Karakteristik *Input-Output* Unit *Thermal*

Karakteristik input-output direpresentasikan dalam persamaan linier yang didapat dari pendekatan atau linearisasi biaya bahan bakar sebagai *input* generator terhadap daya *output* generator yang diperoleh berdasarkan beberapa cara, antara lain:

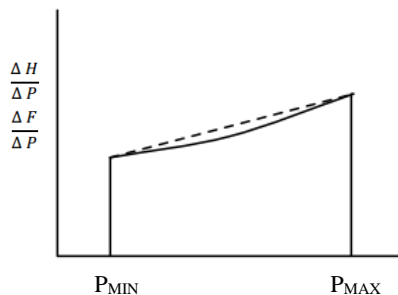
1. Berdasarkan percobaan tentang efisiensi dari pembangkit.
2. Berdasarkan data historis mengenai operasi dari unit generator.
3. Berdasarkan pada data desain dari unit generator yang diberikan oleh pabrik pembuat generator.

Pada Gambar 2.3 karakteristik *input-output* pembangkit di atas terlihat bahwa *input* dari sistem yaitu energi panas yang dibutuhkan per jam dalam $H = (\text{MBtu/h})$ atau biaya total bahan bakar per jam dalam $F(\text{R/h})$, sedangkan *output* dari sistem berupa daya dalam satuan MW. Pada pembangkit termal tersebut terdapat batas daya yang diizinkan dalam pengoperasian yaitu nilai daya minimal (P_{\min}) dan daya maksimal (P_{\max}) [4].

Batas minimum dari daya output generator diperhatikan, hal ini disebabkan karena kondisi teknis atau faktor lain pada *boiler* dan *turbin*. Pada *turbin*, kondisi teknis yang dimaksud adalah kondisi *shell* serta perbedaan suhu *metal rotor*, *exhaust hood temperature*, *rotor expansion* dan *shell expansion*. Pada boiler, kondisi teknis tersebut disebabkan karena *fuel combustion stability and valve*. Sedangkan batas maksimal dari daya output suatu pembangkit ditentukan dari desain kapasitas boiler dan turbin generator [3]. Dengan menggunakan karakteristik *input-output* masing-masing pembangkit, dapat ditentukan pengoperasian optimum (ekonomis) dari sejumlah unit pembangkit dimana total biaya operasinya minimal.

2.3.2 Karakteristik Kenaikan Biaya/Panas Unit Thermal

Karakteristik kenaikan panas dari unit pembangkit uap diperlihatkan pada Gambar 2.4. Karakteristik ini adalah kemiringan dari karakteristik *input-output* ($\Delta H/\Delta P$ atau $\Delta F/\Delta P$). Data yang ditunjukkan pada kurva ini dalam satuan Btu per kilowatt jam (R per kilowatt jam). Lebih lanjutnya, karakteristik ini digunakan untuk perhitungan pembebanan ekonomis dari unit pembangkit. Jika persamaan *input-output* unit pembangkit dinyatakan dalam pendekatan dengan menggunakan persamaan kuadrat, maka karakteristik kenaikan biaya akan mempunyai bentuk garis lurus [4]



Gambar 2. 4 Kurva Karakteristik Kenaikan Biaya

2.4 Economic Dispatch

Economic dispatch (ED) merupakan permasalahan ekonomis utama dalam operasi sistem tenaga listrik. Pembagian pembebanan pada pembangkit-pembangkit yang ada dalam sistem secara optimal dan ekonomi, pada harga beban sistem tertentu dengan menentukan besarnya daya yang dibangkitkan dari masing-masing generator dengan batasan-batasan teknis dan oprasional merupakan tujuan dari *economic dispatch* [saadat]. Batasan terpenting dari operasi sistem pembangkitan daya untuk melayani kebutuhan daya dengan baik adalah total daya yang dibangkitkan harus sama dengan beban daya permintaan yang dilayani oleh sistem ditambah rugi-rugi dari transmisi, agar keandalan sistem dalam melayani kebutuhan daya tetap terpenuhi.

Pada optimisasi yang dilakukan dalam permasalahan ED adalah optimisasi dari segi biaya bahan baku pembangkitan atau *fuel cost* yang memiliki karakteristik kurva biaya tidak linear. Bentuk tipikal dari persamaan biaya pembangkit pembangkit adalah persamaan *polynomial* orde dua dan direpresentasikan Persamaan (2.1). Nilai-nilai karakteristik unit *input-output* pembangkit dapat diperoleh berdasarkan historis dari efisiensi dan operasi unit pembangkit serta berdasarkan pada data desain pembangkit yang diberikan oleh manufaktur.

Batasan dalam permasalahan ED dibagi menjadi dua yaitu *equality constraint* dan *inequality constraint*. *Equality constraint* yaitu pertimbangan dalam perhitungan ED yang menyertakan beberapa faktor eksternal dari generator untuk membangkitkan daya sesuai dengan kebutuhan beban. *Inequality constraint* yaitu pertimbangan dalam perhitungan ED yang menyertakan beberapa faktor internal dari suatu pembangkit dalam mensuplay daya.

Operasi sistem tenaga listrik pada frekuensi konstan dapat disebut juga power balance, yaitu pembangkitan daya real sama dengan total beban ditambah rugi-rugi transmisi. Sehingga kombinasi daya output yang dibangkitkan oleh tiap-tiap generator pada sistem dapat memenuhi kebutuhan daya dari sistem tenaga listrik (*equality constraint*), yang dituliskan sebagai persamaan (2.2).

$$P_G = P_D + P_L \quad (2.2)$$

Jika kerugian transmisi diabaikan, maka persamaan di atas menjadi,

$$P_G = P_D \quad (2.3)$$

Dimana

PG = Total daya dari semua pembangkit
PD = Total daya yang dibutuhkan oleh beban

Batasan yang perlu diperhatikan selanjutnya yaitu daya yang dibangkitkan generator harus memenuhi batas minimal dan maksimal dari unit-unit generator itu sendiri (*inequality constraint*), yang dituliskan sebagai persamaan (2.4):

$$P_{Gi}min \leq P_G \leq P_{Gi}max \quad (2.4)$$

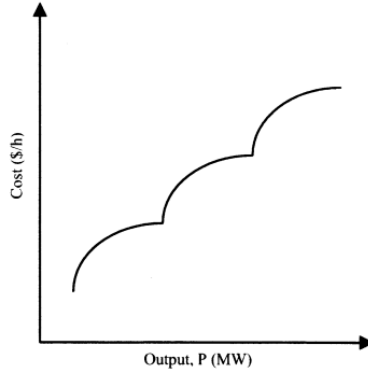
Dengan P_{Gi} adalah besaran daya yang di bangkitkan generator ke-
i. Persamaan (2.4) disebut juga *inequality constrain*.

2.4.1 Valve Point Effect

Pada kenyataannya unit pembangkit dengan katup turbin uap banyak menunjukkan variasi fungsi biaya bahan bakar yang beragam. Sehingga kurva karakteristik dari *output* pembangkit tidak sehalus seperti tampak pada Gambar 2.3 dan Gambar 2.4. Dalam pembangkit termal berskala besar terdapat katup untuk pengaturan uap masuk yang membuka secara bertahap, untuk mendapatkan luaran daya yang semakin meningkat. Efek bukaan katup ini disebut *valve-point effect* (VPE) [6]. VPE menyebabkan ripple pada kurva pembangkitan seperti yang ditunjukkan pada Gambar 3.1. Fungsi biaya menjadi fungsi non-linier dalam orde tinggi. Oleh Karena itu keadaan non-linier ini harus dimasukkan dalam perhitungan untuk mendapatkan perhitungan optimasi dengan ketepatan sebaik mungkin. Kenaikan fungsi biaya dari unit pembangkit dengan efek katup direpresentasikan pada persamaan (3.1) [5].

$$F_i(P_i^t) = a_i + b_i P_i^t + c_i P_i^{2,t} + |e_i \times \sin(f_i \times (P_{min,i}^t - P_i^t))| \quad (2.5)$$

Dimana e_i dan f_i adalah koefisien dari generator i yang merefleksikan efek katup.



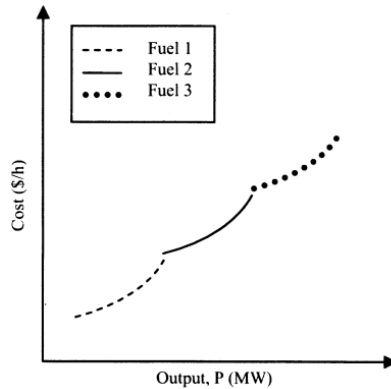
Gambar 2. 5 Kurva Biaya Bahan Bakar dengan 3 Katup

2.4.2 Multiple Fuel Options

Tujuan utama "*Economic dispatch*" dengan opsi beberapa pilihan bahan bakar" adalah untuk menemukan bahan bakar mana yang paling ekonomis untuk dibakar. Katakanlah pabrik yang terdiri dari banyak unit pembangkit, yang dipasok dengan banyak pilihan (Batubara, Gas Alam dan Minyak) dari bahan bakar sehingga sulit dalam penentuan bahan bakar mana yang paling ekonomis untuk digunakan. Fungsi kuadrat *piecewise* digunakan untuk mewakili beberapa pilihan bahan bakar dan fungsi biaya dan biaya tambahan diilustrasikan pada Gambar 3.2. Fungsi biaya *hybrid* dengan batasan ketidaksetaraan direpresentasikan pada persamaan (3.2) [6,7].

$$F_i(P_i^t) = \begin{cases} a_{i1} + b_{i1}P_i^t + c_{i1}P_i^{2,t}, (fuel\ 1) & P_{min,i}^t \leq P_i^t \leq P_{1i}^t \\ a_{i2} + b_{i2}P_i^t + c_{i2}P_i^{2,t}, (fuel\ 2) & P_{1i}^t \leq P_i^t \leq P_{2i}^t \\ \dots \dots \dots & \dots \dots \dots \\ \dots \dots \dots & \dots \dots \dots \\ a_{ik} + b_{ik}P_i^t + c_{ik}P_i^{2,t}, (fuel\ k) & P_{(ik-1)}^t \leq P_i^t \leq P_{max,i}^t \end{cases} \quad (2.6)$$

Dimana a_{ik}, b_{ik}, c_{ik} adalah koefisien biaya dari generator ke-I dengan bahan bakar yang digunakan pada level daya tertentu.



Gambar 2. 6 Kurva Biaya dengan Suplai 3 Type Bahan Bakar

Contoh di Indonesia yaitu PT. Indonesia Power, selaku anak perusahaan dari PT. PLN, mempunyai beberapa pembangkit listrik yang tersebar di Indonesia yang salah satu diantaranya adalah UBP Perak Grati. UBP Perak Grati mengelola PLTGU Grati Blok I (combined cycle) terdiri dari 3 Gas Turbine dan 1 Steam Turbine serta Blok II (open cycle) terdiri dari 3 Gas Turbine dengan desain dual firing menggunakan bahan bakar minyak (HSD) dan bahan bakar gas. PLTGU Grati Blok I dioperasikan untuk kebutuhan base load, sedangkan PLTGU Blok II dioperasikan untuk memenuhi kebutuhan peak load.

PLTDG Pesanggaran terdiri dari empat blok dimana masing-masing blok memiliki tiga unit mesin, didesain menggunakan tiga jenis bahan bakar yaitu HSD (High Speed Diesel), MFO (Marine Fuel Oil), dan Gas. PLTDG yang dikerjakan oleh Consortium Wartsila- PT PP (Persero) ini sudah memulai awal operasi pada 4 April 2015 untuk blok 1 dan beroperasi penuh semua Blok pada 5 Juni 2015.

2.5 Dynamic Economic Dispatch

Dynamic Economic Dispatch (DED) adalah pembagian daya yang harus dibangkitkan oleh generator dalam suatu sistem tenaga listrik secara realtime dalam rentang waktu yang ditentukan (24 jam) sehingga dapat memenuhi kebutuhan beban dengan biaya minimum. Tujuan utama DED adalah meminimalkan konsumsi bahan bakar dari generator pada suatu sistem untuk memperoleh kondisi optimal. Hal ini dikenal dengan istilah optimisasi DED.

Persamaan dasar untuk mendapatkan biaya pembangkitan sama dengan ED (2.1) namun, karena DED melakukan perhitungan secara real time (24 jam) maka persamaan biayanya dimodifikasi menjadi:

$$F_i^t(P_i^t) = a_i P_i^{2t} + b_i P_i^t + c_i \quad (2.7)$$

Pada persamaan (2.5) P_i^t adalah daya yang dibangkitkan dari unit ke- i dalam waktu t . Fungsi objektif tersebut akan meminimalkan biaya yang dibutuhkan oleh masing-masing pembangkit. Berbeda dengan ED biasa, DED memiliki batasan yang disebut *ramp rate*. Nilai dari *ramp rate* terbagi menjadi dua, yaitu *ramp up* dan *ramp down* sesuai dengan karakteristik pada pembangkit yang ditunjukkan pada persamaan (2.6).

$$UR_i \geq P_i^t - P_i^{t-1} \quad (2.8)$$

$$DR_i \geq P_i^{t-1} - P_i^t$$

Variabel waktu dari DED berakibat pada penentuan batas maksimum dan minimum generator tiap waktu akan bervariasi bergantung *output* daya pada waktu sebelumnya dan batasan maksimum dan minimum desain. Sesuai dengan karakteristik batasan *ramp rate* pada pembangkit. Sehingga penentuan batas atas dan batas bawah dari daya yang dibangkitkan oleh generator untuk tiap jam berikutnya dapat dituliskan pada persamaan (2.7).

$$P_i^{min}(t+1) = \max(P_i(t) - DR_i^{max}, P_i^{min}(0)) \quad (2.9)$$

$$P_i^{max}(t+1) = \min(P_i(t) - UR_i^{max}, P_i^{min}(0))$$

Pada tugas akhir ini nilai dari *ramp rate* diasumsikan pada nilai yang konstan [].

2.6 Ant Colony Optimization

Metode ini pertama ditemukan oleh Marco Dorigo (1991) yaitu suatu teknik untuk menentukan jalur yang paling pendek antara titik sumber dengan suatu daerah yang dituju. Pada awalnya semut mencari lintasan secara acak. Ketika semut menemukan sumber makanan, semut akan berjalan kembali ke sarangnya dengan meninggalkan jejak berupa feromon yang menunjukkan jalan menuju sumber makan tersebut. Ketika semut-semut yang lain menemukan jejak feromon tersebut, maka semut-semut yang lain akan berjalan mengikuti jejak feromon menuju sumber makanan. Semut-semut kemudian

mengarahkan koloninya ke jejak feromon tadi. Ketika semut-semut tersebut membawa makanan kembali ke sarang, jejak feromon tadi akan semakin tebal. Jejak feromon akan semakin tebal karena begitu banyak semut yang melewati lintasan yang sama. Jejak feromon dilintasan yang jauh semakin lama akan menguap dan menghilang. Hal ini mulai berhenti ketika ada semut yang menemukan sumber makanan yang lain. Setiap semut adalah agen sederhana yang memiliki karakteristik sebagai berikut:

- a. Semut memilih simpul yang akan dikunjungi dengan mempertimbangan probabilitas. Probabilitas di sini adalah sebuah fungsi yang dipengaruhi oleh jarak simpul dan jumlah feromon yang terkandung dalam sebuah *arc*
- b. Agar semut tidak terjebak dalam path yang sama, kunjungan ke simpul yang sudah dikunjungi tidak diperbolehkan.
- c. Semut akan meninggalkan sebuah trail di sebuah *arc* (i, j) yang dilewatinya.

2.6.1 Graf ACO

Secara garis besar metode *Ant Colony* permasalahan terlebih dahulu dijabarkan dalam bentuk *graft*/jaringan. Dalam *graft* tersebut didefinisikan (N, E) yang merupakan himpunan simpul dan percabangan yang nantinya menjadi jalur semut untuk menuju sumber makanan.

Setiap akan membentuk tour berdasarkan *state transition rule*. Ketika membentuk suatu *tour*, setiap semut akan mengubah besarnya *pheromone* pada cabang yang dilaluinya berdasarkan *local updating rule*. Apabila semua semut telah selesai membentuk suatu *tour*, jumlah *pheromone* pada setiap cabang akan mengalami perubahan sesuai dengan *global updating rule*.

2.6.2 State Transition Rule [9,10]

Pemilihan simpul yang digunakan dalam metode ACO dirumuskan sebagai berikut:

$$s(i, j) = \begin{cases} \arg \max_{j \in U_k} ((\tau_{ij})(\eta_{ij})^\beta), & q \leq q_0 \\ P_{i,j}, & q \geq q_0 \end{cases} \quad (2.10)$$

$$P_{i,j} = \frac{(\tau_{ij})(\eta_{ij})^\beta}{\sum_{j \in U_k} (\tau_{ij})(\eta_{ij})^\beta} \quad (2.11)$$

Pada persamaan (2.8) τ_{ij} adalah *pheromone* pada *arc* (i,j) , η_{ij} adalah *invers* dari pada *arc* (i,j) , U_k merupakan himpunan simpul yang belum dikunjungi semut ke- k yang berada pada simpul i , dan β adalah sebuah parameter yang menentukan hubungan antara *pheromone* dan informasi heuristik dari fungsi biaya dengan nilai ($\beta > 0$). Sedangkan q adalah variabel acak antara 0 dan 1, q_0 parameter yang menentukan tingkat eksploitasi dan eksplorasi ($0 \leq q_0 \leq 1$).

2.6.3 Local Pheromone Update [9]

Setiap kali membentuk *tour*, semut-semut tersebut akan melewati *arc* yang ada dan mengubah besarnya *pheromone*. Hal ini ditunjukkan pada persamaan (2.10).

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \rho\Delta\tau_{ij}, 0 < \rho < 1 \quad (2.12)$$

2.6.4 Global Pheromone Update [9]

Setelah semua semut telah melalui setiap simpul maka besarnya jumlah *pheromone* akan berubah berdasarkan persamaan global updating rule yang ditunjukkan pada persamaan (2.11).

$$\Delta\tau_{ij} = \frac{1}{L_{gb}} \text{ untuk } (i,j) \in \text{global} - \text{best} - \text{tour} \quad (2.13)$$

Dimana L_{gb} adalah panjang dari *tour* terbaik (terbaik) yang dihasilkan oleh sekumpulan semut tersebut.

Halaman ini sengaja dikosongkan

BAB 3

IMPLEMENTASI ANT COLONY OPTIMIZATION PADA PERMASALAHAN DED

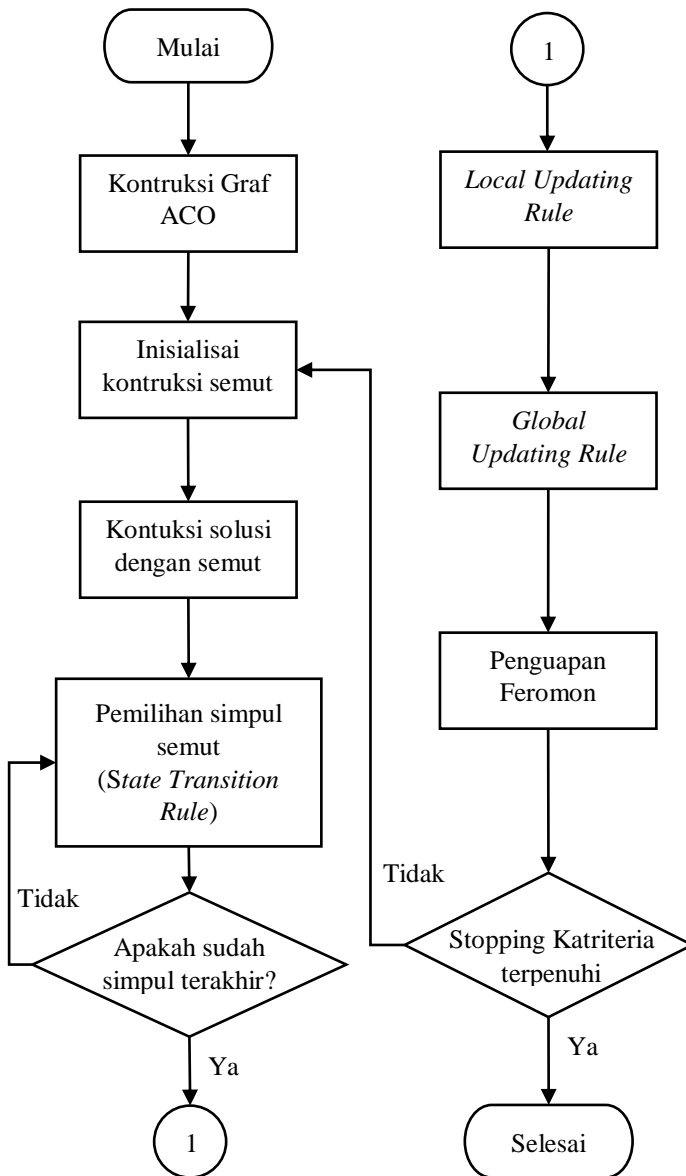
Pada bab ini membahas tentang perancangan serta penerapan algoritma *ant colony optimization* (ACO) untuk permasalahan *Dynamic Economic Dispatch* (DED). Hasil yang diharapkan yaitu biaya pembangkitan paling minimum dari beberapa unit pembangkit thermal.

3.1 Penerapan Algoritma Ant Colony Optimization

Algoritma modified artificial bee colony merupakan metode yang akan digunakan dalam menyelesaikan perhitungan biaya ekonomis pada permasalahan dynamic economic dispatch. Bentuk alur dapat dilihat pada flowchart Gambar 3.1.

Penjelasan flowchart:

1. Hal terpenting pada metode ACO adalah poenjabaran graf. Langkah terlebih dahulu yaitu membangun graf ACO sebagai jalur untuk pencarian solusi.
2. Inisialisasi kontruksi semut yakni parameter ACO yang dibutuhkan dalam pencarian solusi. Seperti intensitas feromon, intensitas penguapan feromon dan jumlah populasi.
3. Proses pencarian solusi dilakukan oleh tiap semut berdasarkan *state transition rule* dang meng update jumlah feromon pada setiap graf yang dilaluinya berdasarkan *local updating rule*, pencarian solusi terus dilakukan *local search*
4. Agen semut memilih simpul yang akan dikunjungi dengan pertimbangan probabilitas. Probabilitas disini adalah sebuah fungsi yang di pengaruhi oleh jarak simpul dan jumlah feromon yang terkandung dalam graf.
5. Semut akan meninggalkan sebuah *trail* pada graf yang dilewati setelah mendapatkan solusi terbaik yang memungkinkan.
6. Melakukan *local search* pada setiap solusi.
7. Meng-update jumlah feromon berdasarkan *global updating rule* pada graf yang membentuk solusi terbaik.



Gambar 3. 1 Flowchart Algoritma ACO

3.2 Inisialisasi Permasalahan DED

Permasalahan DED merupakan permasalahan ED yang berulang dengan rentang waktu 24 jam. Sehingga variabel waktu perlu ditambahkan pada tiap permasalahan ED. Maka permasalahan DED dapat dinotasikan pada sub-bab berikut ini.

3.2.1 Fungsi Objektif

Fungsi objektif dari permasalahan DED merupakan perulangan dari fungsi objektif DED dalam rentang waktu tertentu seperti yang telah dibahas sebelumnya pada persamaan (2.1). Pada dasarnya ada dua macam fungsi biaya yang digunakan pada pengujian tugas akhir ini. Fungsi biaya permasalahan DED dengan memperhatikan *valve point effect* yang ditunjukkan pada persamaan (2.5) dan fungsi biaya permasalahan DED dengan memperhatikan *multiple fuel options* yang ditunjukkan pada persamaan (2.6).

3.2.2 Konstrain Kesetimbangan Daya

DED bekerja di atas horizon waktu sehingga besar daya permintaan berubah-ubah sesuai dengan permintaan daya tiap waktu. Konstrain kesetimbangan daya tanpa mempertimbangkan kerugian daya ditunjukkan pada persamaan (3.1).

$$\sum_{i=1}^N P_i^t = P_D^t \quad (3.1)$$

3.2.3 Konstrain Batasan Oprasional

Setiap unit pembangkit memiliki batasan minimum dan maksimum pembangkitan. Selain itu terdapat satu batasan desain generator yaitu batasan *ramp-rate*. *Ramp-rate* adalah besaran laju respon dari pembangkit atau perubahan daya *output* per satuan waktu. Pada batasan *ramp-rate* juga terdapat laju nilai naik (UR) dan laju nilai turun (DR). Batasan *ramp-rate* digunakan untuk mencegah efek yang tidak diinginkan akibat dinamika perubahan yang cepat dan melebihi kemampuan generator, baik dalam pembebanan maupun pelepasan daya. Batasan ini akan menjadi penting saat dilakukan penjadwalan generator dalam satu periode waktu. Sehingga akan mempengaruhi batasan unit generator untuk waktu berikutnya [3]. Untuk lebih jelasnya Batasan oprasi dengan *ramp-rate* ditunjukkan pada persamaan (3.2).

$$\begin{aligned} P_{min,i}^t &= \max(P_{min,i}, P_i^{t-1} - DR_i) \\ P_{max,i}^t &= \min(P_{max,i}, P_i^{t-1} + UR_i) \end{aligned} \quad (3.2)$$

3.3 Perancangan *Plant*

Pada tugas akhir ini menggunakan dua *plant* yang akan digunakan untuk pengujian perhitungan *dynamic economic dispatch*. Data pengujian pertama yang digunakan yaitu berdasarkan jurnal IEEE sistem tenaga listrik 5 unit pembangkit dengan pertimbangan *valve point effect* menggunakan metode *Ant Colony Optimization* [lagrang]. Data pengujian kedua yang digunakan yaitu berdasarkan jurnal IEEE sistem tenaga listrik 10 unit pembangkit dengan pertimbangan *valve point effect* dan *multiple fuel options* [11] data mengenai koefisien biaya dan kapasitas pembangkit akan dibahas pada sub bab selanjutnya.

3.3.1 Sistem 5 Unit Pembangkit

Plan ke-1 data yang digunakan yaitu sistem kelistrikan 5 Unit pembangkit. Berikut ini adalah data fungsi biaya 5 Unit pembangkit ditunjukkan pada Tabel 3.1. Fungsi biaya tersebut telah disertakan dengan efek dari bukaan katup yang direpresentasikan dalam *cost coefficient*. Karakteristik dan kemampuan dari masing-masing generator yang beroperasi ditunjukkan pada Tabel 3.2 serta data beban ditunjukkan pada Tabel 3.2 dalam satuan MW.

Tabel 3. 1 Data Fungsi Biaya Sistem 5 Unit Pembangkit

Unit	Cost Function				
	a_i	b_i	c_i	e_i	f_i
1	0.0080	2.0	25	100	0.042
2	0.0030	1.8	60	140	0.040
3	0.0012	2.1	100	160	0.038
4	0.0010	2.0	120	180	0.037
5	0.0035	1.8	40	200	0.015

Tabel 3. 2 Data Generator Sistem 5 Unit Pembangkit

Unit	P min (MW)	P max (MW)	UR (MW/h)	DR (MW/h)
1	10	75	30	30
2	20	125	30	30
3	30	175	40	40
4	40	250	50	50
5	50	300	50	50

Tabel 3. 3 Data Generator Sistem 5 Unit Pembangkit

Jam	Total Beban (MW)	Jam	Total Beban (MW)
1	410	13	704
2	435	14	690
3	475	15	654
4	530	16	580
5	558	17	558
6	608	18	608
7	626	19	654
8	654	20	704
9	690	21	680
10	704	22	605
11	720	23	527
12	740	24	463

3.3.2 Sistem 10 Unit Pembangkit

Plan ke-2 data yang digunakan yaitu sistem kelistrikan 10 Unit pembangkit. Pada *plant* ke-2 ketika MFO masuk dalam pertimbangan, diasumsikan setiap 1 area unit pembangkit terdapat 3 jenis generator dengan biaya bahan bakar yang berbeda [11]. Untuk batas atas dan batas bawah operasi generator serta fungsi biaya yang digunakan dari tiap generator direpresentasikan dalam persamaan (3.3) dan (3.4).

$$\begin{aligned}
 P_{1,i}^{LMOF} &= P_i^{min} ; P_{1,i}^{UMFO} = P_i^{min} + 0.333 (P_i^{max} - P_i^{min}) \\
 P_{2,i}^{LMOF} &= P_{1,i}^{UMFO} ; P_{2,i}^{UMFO} = P_i^{min} + 0.667 (P_i^{max} - P_i^{min}) \\
 P_{3,i}^{LMOF} &= P_{2,i}^{UMFO} ; P_{3,i}^{UMFO} = P_i^{min}
 \end{aligned} \tag{3.3}$$

$$\begin{aligned}
 a_{1,i} &= a_i ; b_{1,i} = b_i ; c_{1,i} = c_i \\
 a_{2,i} &= 1.1 a_i ; b_{2,i} = 1.1 b_i ; c_{2,i} = 1.1 c_i \\
 a_{3,i} &= 1.2 a_i ; b_{3,i} = 1.2 b_i ; c_{3,i} = 1.2 c_i
 \end{aligned} \tag{3.4}$$

Berikut ini adalah data fungsi biaya 10 Unit pembangkit ditunjukkan pada Tabel 3.4. Karakteristik dari masing-masing generator yang beroperasi ditunjukkan pada Tabel 3.5 dan data beban ditunjukkan pada Tabel 3.2 dalam satuan MW.

Tabel 3. 4 Data Fungsi Biaya Sistem 10 Unit Pembangkit

Unit	Cost Function				
	a _i	b _i	c _i	e _i	f _i
1	0.00063	21.05	1313.6	600	0.036
2	0.00043	21.60	958.5	450	0.041
3	0.00070	23.90	471.6	260	0.052
4	0.00039	20.81	604.97	320	0.028
5	0.00079	21.62	480.29	280	0.063
6	0.00056	17.87	601.75	310	0.048
7	0.00480	23.23	639.4	340	0.082
8	0.10908	19.58	455.6	270	0.098
9	0.00211	16.51	502.7	300	0.086
10	0.00951	22.54	692.4	380	0.094

Tabel 3. 5 Data Generator Sistem 10 Unit Pembangkit

Unit	P min (MW)	P max (MW)	UR (MW/h)	DR (MW/h)
1	135	460	80	80
2	150	470	80	80
3	60	300	80	80
4	73	340	50	50
5	73	243	50	50
6	57	160	50	50
7	47	120	50	50
8	20	80	30	30
9	20	130	30	30
10	55	55	30	30

Tabel 3. 6 Data Beban Sistem 10 Unit pembangkit

Jam	Total Beban (MW)	Jam	Total Beban (MW)
1	1036	13	2072
2	1110	14	1924
3	1258	15	1776
4	1406	16	1554

Tabel 3. 7 Data Beban Sistem 10 Unit Pembangkit (lanjutan)

Jam	Total Beban (MW)	Jam	Total Beban (MW)
5	1480	17	1480
6	1628	18	1628
7	1702	19	1776
8	1776	20	2072
9	1924	21	1924
10	2072	22	1628
11	2146	23	1332
12	2220	24	1184

3.4 Algoritma ACO untuk Permasalahan DED

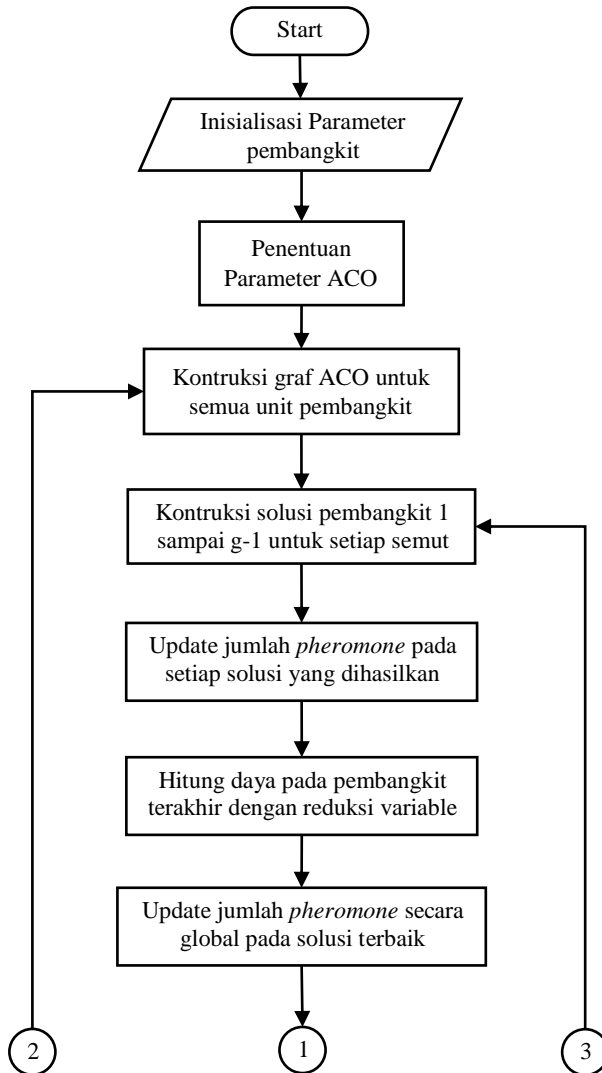
Pada sub bab ini akan dijelaskan penerapan algoritma *Ant Colony Optimization* dalam menyelesaikan perhitungan biaya ekonomis pada permasalahan *dynamic economic dispatch* (DED). Pada Gambar 3.3 terlihat diagram alir penyelesaian DED menggunakan metode *Ant Colony Optimization* (ACO). ACO akan menentukan kombinasi daya output dari tiap unit pembangkit dengan biaya yang paling ekonomis untuk memenuhi permintaan beban dalam periode waktu.

3.5 Parameter Ant Colony Optimization

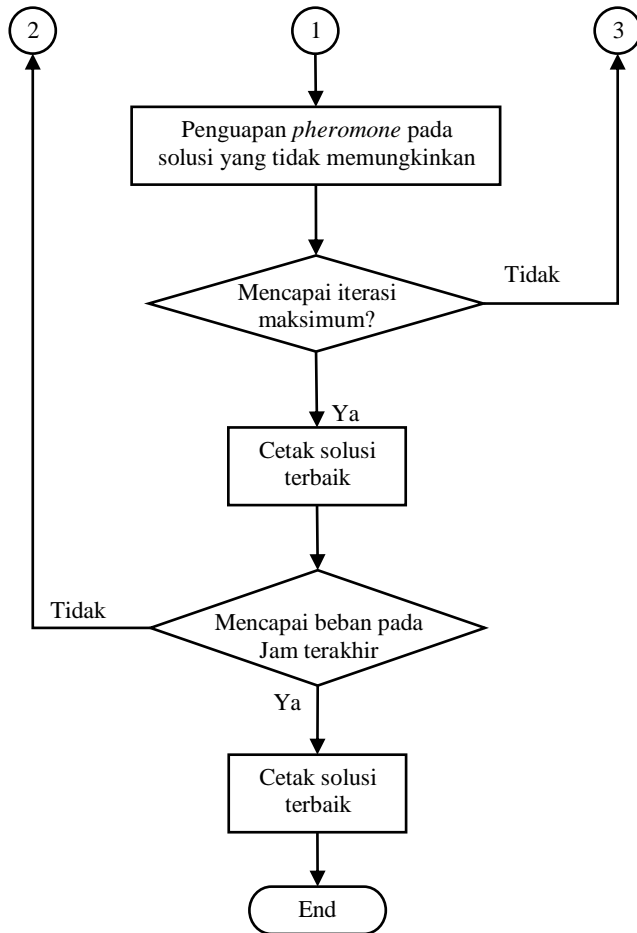
Parameter yang digunakan pada algoritma ACO untuk menyelesaikan permasalahan DED terdiri dari

Tabel 3. 8 Tabel Korelasi Parameter ACO

Parameter	Keterangan
Graf ACO (n)	Berupa vector dari kenaikan level daya pembangkit dari Pmin – Pmax berdasarkan interval yang ditentukan.
Feromone (τ)	Berupa nilai daya yang berada pada jalur Graf.
Individu semut (m)	Variable yang menginisialisasikan fungsi biaya dengan batasan <i>inequality</i> dan <i>equality constrain</i> .
(β)	Sebuah hubungan parameter mengenai jarak (biaya) dan feromon (daya).
(ρ)	Intensitas penguapan pada feromon



Gambar 3. 2 Flowchart Implementasi ACO Pada DED



Gambar 3. 3 Flowchart Implementasi ACO Pada DED Lanjutan

Penjelasan *flowchart*:

1. Inisialisasi input program dan input dari *plant* yang akan dioptimalkan. Input program dala algoritma *Ant Colony Optimization* yang nantinya dijadikan suatu parameter yaitu:

τ_0 = besar intensitas feromon mula-mula pada tiap simpul yang dilewati oleh agen semut.

n = jumlah jalur percabangan yang akan dilalui oleh semut

ρ = koefisien dalam menentukan tingkat penguapan feromon.

Ite = banyaknya iterasi maksimal yang dilakukan.

β = tingkat relative informasi heuristik.

m = jumlah agen semut.

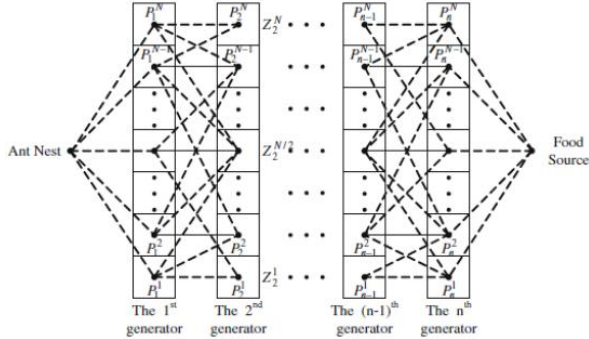
q = penentuan agen semut secara random

Penentuan agen semut dibedakan menjadi dua berdasarkan *state transition rule*, yaitu semut eksplorasi dan semut eksploitasi. Sedangkan input dari *plant* adalah banyaknya unit generator, nilai *cost function*, Batasan oprasional dari generator dan totak beban yang diminta secara periodik.

2. Pada tahap penjabaran graf ACO yang digunakan sebagai batas bawah adalah P_{min} dan sebagai batas atas adalah P_{max} dari masing-masing pembangkit. Untuk menjabarkan permasalahan DED kedalam graf ACO, dibuat interval titik sebanyak N level daya dari batas bawah sampai batas atas generator sesuai pada persamaan (3.5).

$$P_i^j = P_i^{min} + (j - 1) \left(\frac{P_i^{max} - P_i^{min}}{N - 1} \right) \quad (3.5)$$

Interval titik yang dibuat adalah sama untuk masing-masing pembangkit. seperti yang terlihat pada Gambar 3.4. Graf ACO yang telah dibuat sebagai jalur yang nantinya dilewati oleh tiap agen semut.



Gambar 3. 4 Graf ACO

Tiap interval titik pada persamaan (3.6) yang telah dibuat untuk masing-masing pembangkit, ditentukan pula harga pembangkitan sesuai dengan fungsi objektif yang digunakan.

$$titik = \begin{bmatrix} P_1^n & P_2^n & P_3^n & \dots & \dots & P_{g-1}^n & P_g^n \\ P_1^{n-1} & P_1^{n-1} & P_1^{n-1} & \dots & \dots & P_{g-1}^n & P_g^n \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ P_1^2 & P_2^2 & P_3^2 & \dots & \dots & P_{g-1}^2 & P_g^2 \\ P_1^1 & P_2^1 & P_3^1 & \dots & \dots & P_{g-1}^1 & P_g^1 \end{bmatrix} \quad (3.6)$$

3. Pada tahap selanjutnya setiap semut membangun solusi (rute perjalanan) berdasarkan *state transition rule*, dan memperbarui intensitas *pheromone* pada setiap percabangan yang dialulinya. Pencarian solusi dilakukan mulai dari pembangkit ke-1 sampai pembangkit ke g-1
4. Proses pencarian solusi terbaik dengan perhitungan minimum dan maksimum dari batas oprqasi generator berdasarkan aturan *Max-Min Ant System*. Nilai *pheromone* hanya di diperbarui oleh semut yang membangun lintasan terbaik berdasarkan *local updating rule*.
5. Setelah meninggalkan lokasi makanan, agen semut menyimpan memori yang berisi interval titik pembangkitan harga minimum yang memungkinkan dan meninggalkan jejak berupa nilai feromon yang telah diperbarui interval titik tersebut berdasarkan *memory* yang diperoleh.
6. Pada pembangkit yang terakhir dilakukan perhitungan *reduksi variable* dengan cara beban yang diminta dikurangi dengan penjumlahan daya dari pembangkit 1 sampai pembangkit g-1 sesuai pada persamaan (3.7).

$$P_g = P_D - \sum_{i=1}^{g-1} P_i \quad (3.7)$$

7. Tahap berikutnya memperbarui nilai *pheromone* untuk stiap iterasi berdasarkan *global updating rule* pada masing-masing simpul graf yang dibuat.
8. Penguapan *pheromone* pada jalur yang telah dilewati agen semut sebelumnya dengan solusi yang tidak memungkinkan.
9. Menampilkan hasil yang didapat.

3.6 Kontruksi Solusi yang Memungkinkan

Kontruksi solusi yang memungkinkan dilakukan oleh semut dengan cara memilih satu level daya yang sudah didiskritisasi ke dalam graf ACO mulai dari pembangkit 1 sampai pembangkit g-1.

Dalam mengkonstruksi solusi yang memungkinkan dari pembangkit 1 sampai pembangkit g-1 ini dilakukan dengan mengikuti aturan *State Transition Rule* dan aturan *Mini Maxi Ant System*, sementara untuk pemilihan daya pembangkit sepuluh dilakukan dengan aturan *reduksi variable*.

3.6.1 Pemilihan Solusi Terbaik

Pemilihan solusi terbaik dilakukan berdasarkan aturan dari algoritma ACO yaitu *state transition rule*. Pada aturan *state transition rule* setiap semut membangun solusi (rute perjalanan) dan memperbarui intensitas *pheromone* pada setiap percabangan yang dialulinya. Terdapat dua cara untuk semut dalam membangun kontruksi yaitu eksploitasi dan eksplorasi yang ditentukan secara random dengan batasan q_0 . Sesuai dengan persamaan (3.8), dimana P_i^j adalah simpul yang akan dipilih oleh semut dan simpul tersebut adalah level daya ke-j dari pembangkit ke-i.

$$P_i^j = \begin{cases} \max \{ \tau_{i,j} [\eta_{i,j}]^\beta \}, & \text{bila } q \leq q_0 \\ \max \frac{\{ \tau_{i,j} [\eta_{i,j}]^\beta \}}{\sum_{m=0}^{m_i} \{ \tau_{i,m} [\eta_{i,m}]^\beta \}}, & \text{bila } q \geq q_0 \end{cases} \quad (3.8)$$

Dari persamaan (3.10) τ_{ij} adalah intensitas feromon dan η_{ij} adalah informasi heuristic untuk simpul P_i^j . Selanjutnya $\eta_{i,j}$ adalah parameter informasi heuristic dari biaya minimum yang memungkinkan dan direpresentasikan $\eta_{i,j}$ dalam Persamaan (3.9).

$$\eta_{i,j} = \frac{1}{F_{ij}} \quad (3.9)$$

F_{ij} adalah harga minimum yang memungkinkan dipilih oleh tiap semut dari suatu simpul (i, j) .

Setelah melakukan perjalanan untuk mencari solusi terbaik, setiap semut akan memperbarui intensitas *pheromone* pada jejak yang telah dilalui berdasarkan aturan *local update pheromone*. Besarnya intensitas *pheromone* yang diubah ditentukan dengan persamaan (2.12). Sedangkan intensitas *pheromone* pada jalur yang tidak memungkinkan akan mengalami penguapan berdasarkan parameter yang telah ditentukan sebelumnya yakni $(1 - \rho)$.

Setiap iterasi yang dihitung berdasarkan jumlah populasi semut yang telah ditentukan. Jadi setelah semua populasi semut melakukan kontruksi solusei terbaik selanjutnya *pheromone* yang ditinggalkan

semua semut untuk semua kombinasi simpul (i, j) diupdate kembali berdasarkan aturan *global updating rule*. Prinsipnya ketika semua semut telah memilih simpul (i, j) , maka pada simpul tersebut akan terdapat jumlah *pheromone* yang berbanding lurus dengan jumlah semut yang memilih simpul tersebut sesuai pada persamaan (3.10)

$$\Delta\tau_{i,j} = \sum_{k=1}^m \Delta\tau_{i,j}^k \quad (3.10)$$

$\Delta\tau_{i,j}$ didefinisikan sebagai feromon yang ditinggalkan oleh semua semut untuk semua kombinasi.

3.7 Reduksi Variabel

Ketika semut sudah berhasil mengkonstruksi solusi untuk pembangkit 1 sampai pembangkit $g-1$, maka untuk mencari nilai daya pada pembangkit terakhir dilakukan aturan *reduksi variable* yaitu dengan cara beban yang diminta dikurangi dengan penjumlahan daya dari pembangkit 1 sampai pembangkit $g-1$ berdasarkan persamaan (3.7). Setelah nilai daya pada pembangkit terakhir diperoleh, maka nilai tersebut perlu didiskritisasi agar nilai tersebut dapat masuk ke dalam matriks titik. Aturan untuk pendiskritan nilai tersebut direpresentasikan oleh persamaan (3.11). *Nodelast* adalah simpul yang terpilih untuk pembangkit terakhir. P_g adalah daya yang harus disuplai mencapai beban yang ditentukan dan $P_g^{max} - P_g^{min}$ adalah batas atas dan batas bawah dari pembangkit terakhir.

$$nodelast = fix \left(\frac{(P_g - P_g^{min}) * (N - 1)}{P_g^{max} - P_g^{min}} \right) \quad (3.11)$$

3.8 Aturan Mini-Maxi Ant System

Penentuan kemungkinan solusi terbaik oleh tiap agen semut tidak menjamin jika hanya berbekal *state transition rule* dan *global updating rule*. Penentuan solusi terbaik yang memungkinkan harus mematuhi konstarain keseimbangan daya dan batasan oprasi. Pada permasalahan tersebut untuk menaggulangnya yaitu dengan cara memberi nilai feromon yang sangat kecil pada simpul yang nantinya menghasilkan solusi yang tidak memungkinkan. Jadi ketika agen semut bertemu dengan simpul yang tidak memungkinkan dalam memilih solusi pada pembangkit ke- g , pada simpul tersebut akan diberikan nilai intensitas feromon ($\tau = -1$) sehingga semut akan mencoba untuk memilih simpul yang lain. Proses ini juga

bertujuan agar semut tidak terjebak pada simpul yang sama. Maka persamaan (3.12) dan (3.13) perlu disertakan pada *state transition rule*.

$$\begin{aligned} \text{maxi} &\leq \text{beban} - \sum_{i=1}^{k=g} P_k - \sum_{i=1}^{k=g} P_{g-k}^{\text{max}} \\ \text{mini} &\geq \text{beban} - \sum_{i=1}^{k=g} P_k - \sum_{i=1}^{k=g} P_{g-k}^{\text{min}} \end{aligned} \quad (3.13)$$

Beban merupakan daya minimal yang bisa disuplai untuk semua pembangkit, sementara k adalah himpunan dari pembangkit yang sudah terpilih salah satu simpulnya oleh semut. Selanjutnya g adalah himpunan.

Pada permasalahan DED aturan *mini* bertujuan untuk mencegah semut mengunjungi simpul level daya yang terlalu besar. Bila dimisalkan beban yang diminta pada suatu jam adalah 500 MW dan simpul level daya yang dipilih oleh pembangkit 1 adalah 90 MW, sementara penjumlahan daya minimum untuk pembangkit kedua sampai pembangkit terakhir adalah 480 MW. Jika kasus seperti ini terjadi maka ada dua kemungkinan yaitu nantinya suplai daya yang jauh melebihi beban atau salah satu dari pembangkit 2 sampai pembangkit terakhir akan ada yang menyuplai di bawah batas minimumnya.

3.9 Penyimpanan Memory Semut

Penyimpanan memory semut digunakan ketika salah satu simpul level daya dari pembangkit tertentu telah melewati Aturan Mini-Maxi, maka simpul level daya tersebut akan dimasukkan ke dalam memori semut. Memori semut di sini berbentuk matriks. Jika dimisalkan jumlah semut adalah m dan jumlah pembangkit adalah g, maka memori semut direpresentasikan oleh matriks berukuran $[m \times (g-1)]$.

BAB 4

SIMULASI DAN ANALISA

Pada bab ini membahas mengenai pengujian algoritma *ant colony optimization*, (ACO) untuk permasalahan DED. Program yang telah dibuat disimulasikan menggunakan 2 *plant* dengan data yang telah dibahas sebelumnya. Simulasi yang pertama yaitu pada *plant* 5 Unit pembangkit tanpa mempertimbangkan *valve point effect*, (VPE). Simulasi yang kedua mesing menggunakan *plant* yang sama namun dengan pertimbangan *valve point effect* kemudian dari kedua simulasi dilakukan analisa untuk keduanya. *Plant* yang berikutnya yaitu simulasi dengan 10 Unit pembangkit. Simulasi yang ketiga dilakukan pada *plant* 10 Unit pembangkit tanpa mempertimbangkan *valve point effect* dan *multiple fuel options*, (MFO). Simulasi yang keempat menggunakan *plant* yang sama namun dengan pertimbangan *valve point effect* dan *multiple fuel options* kemudian dilakukan analisa untuk keduanya.

Pada algoritma ACO, terdapat 4 macam parameter yang dapat dioptimalkan yaitu: jumlah semut (m), koefisien tingkat penguapan feromon (ρ), intensitas feromon (τ), jumlah simpul dalam penjabaran graf (n), dan tingkat relatif informasi heuristik (β). Untuk setiap kasus bisa jadi memiliki parameter optimal yang berbeda-beda.

4.1 Validasi Program ACO

Sebelum dilakukan simulasi terhadap *plant* yang digunakan terlebih dahulu dilakukan validasi program *ant colony optimization* pada referensi buku Allan J. Wood contoh *economic dispatch* pada *example* 3A dan 3B.

Tabel 4. 1 Data Pembangkit Example 3A.

Pembangkit	Daya (MW)	
	Minimum	Maximum
1	150	600
2	100	400
3	50	200

Tabel 4. 2 Data Fungsi Biaya Example 3A.

Pembangkit	Fungsi Biaya		
	a_i	b_i	c_i
1	0.001562	7.92	561
2	0.001940	7.85	310
3	0.004830	7.97	78

Beban yang digunakan yaitu 850 MW. Sesuai dengan contoh buku dengan iterasi lamda, daya yang dibangkitkan adalah sebagai berikut.

Pembangkit 1 = 392.2 MW

Pembangkit 2 = 334.6 MW

Pembangkit 3 = 122.2 MW

Sehingga biaya total yang dihasilkan adalah = **8194.3561 \$/h**. Sedangkan jika menggunakan metode ACO didapat nilai daya yang dibangkitkan setiap unit pembangkit adalah sebagai berikut:

Pembangkit 1 = 393.4 MW

Pembangkit 2 = 334.4 MW

Pembangkit 3 = 122.2 MW

Sehingga biaya total yang dihasilkan adalah = **8195.195 \$/h**. penerapan metode *Ant Colony Optimization* pada permasalahan *economic dispatch example 3A* buku Allan J. Wood, jika dibandingkan hasilnya dengan contoh permasalahan *example 3A* menggunakan iterasi lambda yang sudah diberikan pada buku, menunjukan perbedaan hasil total biaya operasi dari ketiga unit pembangkit yang tidak signifikan. Pada *example 3B* dengan beban yang sama nilai fungsi biaya berubah pada unit 1 menjadi:

$$F_1(P_1) = (459 + 6.48 P_1 + 0.00128 P_1^2)$$

Perubahan ini menyebabkan perhitungan dengan satu akli iterasi lamda menyebabkan pembangkitan pada unit 1 melebihi batas maksimu. Maka dilakukan perhitungan lebih dari sekali iterasi lamda sampai memenuhi dari batas operasi generator. Sesuai dengan contoh buku dengan iterasi lamda, daya yang dibangkitkan adalah sebagai berikut.

Pembangkit 1 = 600 MW

Pembangkit 2 = 187.1 MW

Pembangkit 3 = 62.9 MW

Sehingga biaya total yang dihasilkan adalah = **7252.83 \$/h**. Sedangkan jika menggunakan metode ACO didapat nilai daya yang dibangkitkan setiap unit pembangkit adalah sebagai berikut:

Pembangkit 1 = 600MW

Pembangkit 2 = 187.1741 MW

Pembangkit 3 = 62.9259 MW

Sehingga biaya total yang dihasilkan adalah = **7253.7 \$/h**. Dapat disimpulkan bahwa metode ACO yang akan digunakan pada penelitian ini dapat digunakan dalam permasalahan *economic dispatch* maupun *dynamic economic dispatch*.

4.2 Sistem Tenaga Listrik 5 IEEE Unit Pembangkit

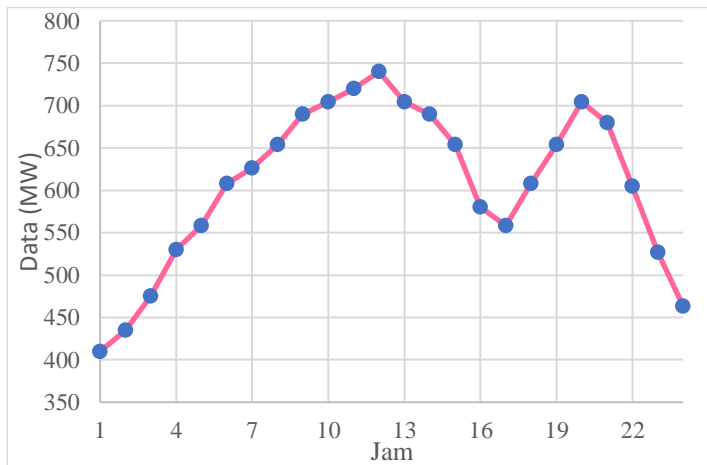
Pada *plant* yang pertama akan dilakukan dua studi kasus simulasi seperti yang sudah dijelaskan sebelumnya.

4.2.1 Simulasi Kasus I Plant I

Pada kasus I ini akan disimulasikan DED tanpa batasan *valve point effect*. Tujuannya agar dapat dibandingkan dengan kasus II bagaimana pengaruh dari *valve point effect* jika dimasukkan kedalam fungsi objektif. Parameter ACO yang digunakan yaitu:

1. Populasi semut (m) sebesar 30
2. Tingkat penguapan feromon (ρ) 0.9
3. Tingkat relatif informasi heuristik (τ) sebesar 0.7
4. Jumlah simpul dalam penjabaran graf (n) sebesar 200
5. Intensitas feromon mula-mula (β) sebesar 0.8

Simulasi kasus I akan dilakukan sesuai dengan permintaan beban selama 24 jam dengan total beban 14.577 MW yang ditunjukkan pada Gambar 4.1. Berikut adalah hasil simulasi DED kasus I yang ditunjukkan pada Tabel 4.3.

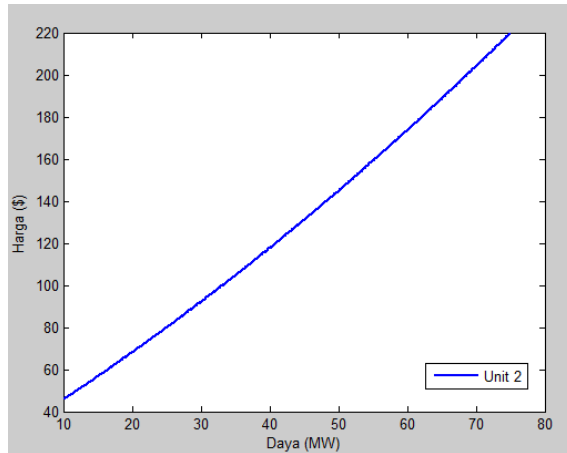


Gambar 4. 1 Kurva Permintaan Beban 24 Jam

Tabel 4. 3 Hasil Simulasi DED 5 Unit Pembangkit

Jam	Unit					Biaya	Daya
	P1	P2	P3	P4	P5		
1	29.5	51.6	73.6	103.1	152.2	\$ 1,198.50	410.0
2	18.8	72.4	65.2	133.5	145.1	\$ 1,251.10	435.0
3	19.3	87.0	85.4	127.6	155.8	\$ 1,342.40	475.0
4	31.5	91.2	74.1	160.1	173.1	\$ 1,469.80	530.0
5	22.7	101.2	87.0	180.1	167.0	\$ 1,534.40	558.0
6	39.3	109.8	100.1	178.0	181.0	\$ 1,654.60	608.0
7	37.0	111.1	114.0	161.0	203.1	\$ 1,697.30	626.0
8	18.2	105.2	93.7	204.6	232.5	\$ 1,764.50	654.0
9	26.5	100.7	90.9	239.9	232.0	\$ 1,851.80	690.0
10	27.7	117.2	130.2	216.3	212.6	\$ 1,882.40	704.0
11	27.8	106.9	129.8	238.8	216.7	\$ 1,921.40	720.1
12	28.2	107.6	149.2	232.0	223.0	\$ 1,970.00	740.1
13	31.5	103.1	126.4	228.7	214.3	\$ 1,882.60	704.1
14	39.0	103.8	133.9	212.6	200.8	\$ 1,849.30	690.1
15	33.2	94.2	112.2	222.5	192.1	\$ 1,762.60	654.0
16	22.8	87.3	141.9	183.5	144.5	\$ 1,589.00	580.0
17	16.4	71.5	105.3	207.9	157.0	\$ 1,537.00	558.0
18	19.8	101.1	82.0	223.8	181.4	\$ 1,654.00	608.0
19	24.6	77.8	113.6	227.7	210.3	\$ 1,763.50	654.0
20	22.4	107.7	151.7	211.2	211.0	\$ 1,882.20	704.0
21	26.5	95.7	118.2	239.1	200.5	\$ 1,824.90	680.0
22	27.6	83.5	134.9	190.8	168.1	\$ 1,646.00	605.1
23	23.0	84.4	146.9	152.1	120.7	\$ 1,469.30	537.1
24	31.4	113.8	122.6	119.9	75.4	\$ 1,333.30	463.0
Total						\$ 39,732.10	14578.6

Tabel 4.3 menunjukkan hasil simulasi DED namun tidak melibatkan *valve point effect* sehingga kurva pembangkitan tiap generator berbentuk non-linier dengan orde dua seperti pada Gambar 4.2 yaitu kurva pembangkitan pada generator 2.

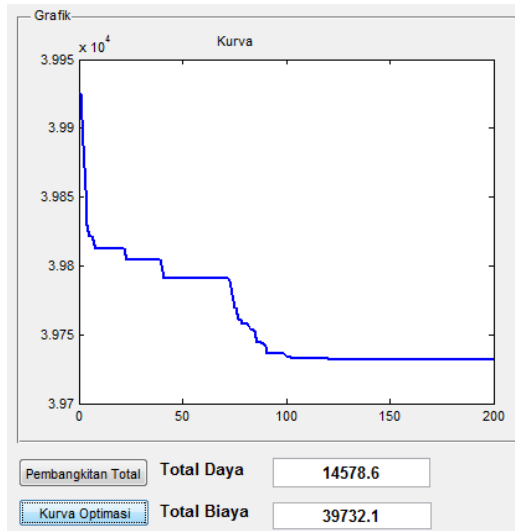


Gambar 4. 2 Kurva Pembangkitan pada Generator 2

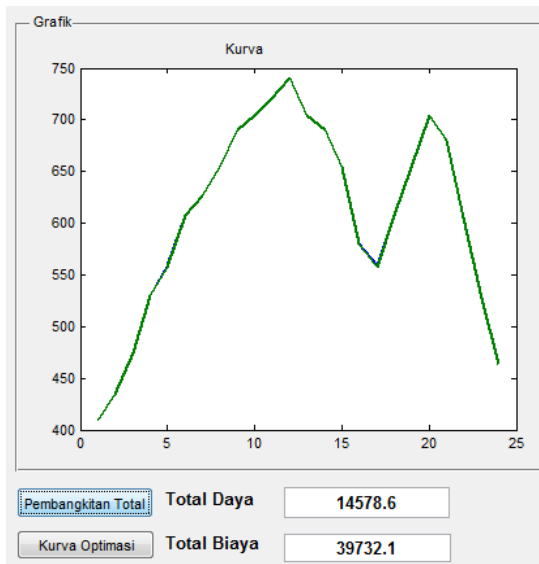
Sistem tenaga listrik dengan 5 Unit pembangkit ini memiliki beban maksimal dan minimal yang mampu disuplai oleh semua pembangkit tanpa melebihi batas oprasi generator. Biaya total pembangkitan \$ **39,732.10** dan daya yang dibangkitkan setiap jamnya tidak kurang dari permintaan beban tiap jam dalam satu periode dengan total daya terbangkitkan **14578.6 MW**.

Artinya total daya yang terbangkitkan memenuhi permintaan beban. Sedangkan batasan oprasi dengan mempertimbangkan *ramp rate* seperti terlihat pada **Unit 1 jam ke-7** daya yang dibangkitkan sebesar **37.0 MW**. Pada **jam ke-8** daya yang dibangkitkan sebesar **18.2 MW**. Maka selisih daya terbangkitkan adalah **18.8 MW** dan batasan *ramp rate* untuk **Unit 1** adalah **30** artinya daya yang dibangkitkan setiap jam oleh masing-masing pembangkit masih dalam batasan *ramp rate*. Hal ini menunjukkan bahwa hasil simulasi sudah memenuhi batasan yang digunakan, seperti *equality constrain* ($P_d = P_g$), *inequality constrain* (P_{min} dan P_{max}) serta *ramp-rate* generator.

Hasil simulasi DED menggunakan metode ACO untuk mendapatkan biaya minimum pembangkitan ditunjukkan pada Gambar 4.3 kurva konvergensi hasil *running* DED dengan iterasi 200. Gambar 4.3 menunjukkan perhitungan pada iterasi ke 100 sudah menunjukkan konvergensinya dengan lama komputasi yakni **344.87 detik**. Gambar 4.4 menunjukkan kurva kestimbangan daya antara pembangkitan dan permintaan beban. Berdasarkan data yang telah disediakan daya yang dibangkitkan lebih besar atau sama dengan daya yang diminta.



Gambar 4. 3 Kurva Konvergensi IEEE 5 Unit Generator.



Gambar 4. 4 Kurva Kestimbangan Daya Pembangkitan dan Pembebanan

Metode ACO dikatan bekerja dengan baik untuk sistem IEEE 5 Unit Generator seperti yang terlihat pada Tabel 4.4 sebagai komparasi dengan metode pembanding lainnya yaitu *Bee Colony Opimization* (ABC). Hasil yang didapat, dengan menggunakan metode ACO biaya pembangkitan sudah mendapatkan hasil yang optimal untuk sistem IEEE 5 Unit Pembangkit dengan waktu komputasi lebih cepat daripada metode ABC.

Tabel 4. 4 Komparasi Metode untuk 5 Unit Pembangkit

Metode	Total Biaya	Komputasi (s)
ABC	\$ 39,660.27	556.89 s
ACO	\$ 39,786.50	344.87 s

4.2.2 Simulasi Kasus II Plant I

Pada kasus II ini akan disimulasikan DED dengan batasan *valve point effect*. Kemudian dibandingkan hasil biaya pembangkitan dengan kasus I untuk mengetahui bagaimana pengaruh dari penambahan *coefficient* dari *valve point effek* pada fungsi objektif yang digunakan. Simulasi kasus II akan dilakukan pembebanan yang sama pada kasus I selama 24 jam. Parameter ACO sengaja tidak dirubah karena telah menemukan hasil yang optimal serta mempermudah dalam analisa. Berikut adalah hasil simulasi DED kasus II yang ditunjukkan pada Tabel 4.5.

Tabel 4. 5 Hasil Simulasi DED 5 Unit Pembangkit dengan VPE

Jam	Unit					Biaya	Daya
	P1	P2	P3	P4	P5		
1	20.0	36.2	52.4	72.4	229.4	\$1,262.00	410.4
2	16.2	59.4	83.2	96.6	179.6	\$1,350.00	435.2
3	33.4	31.1	96.3	94.5	219.7	\$1,436.60	475.0
4	40.5	59.1	134.7	125.9	169.9	\$1,613.80	530.1
5	19.8	87.9	104.2	142.4	203.9	\$1,661.70	558.1
6	16.1	99.5	121.1	128.9	242.5	\$1,751.70	608.1
7	35.3	109.2	138.3	148.2	195.1	\$1,829.40	626.0
8	18.5	86.3	106.0	198.2	245.1	\$1,913.00	654.1
9	15.9	83.6	68.7	237.0	284.9	\$1,942.60	690.1
10	15.5	113.6	54.5	220.5	300.0	\$2,128.10	704.0
11	15.5	110.1	82.2	248.2	264.0	\$2,125.20	720.2
12	31.9	108.3	49.9	250.0	300.0	\$2,082.50	740.0

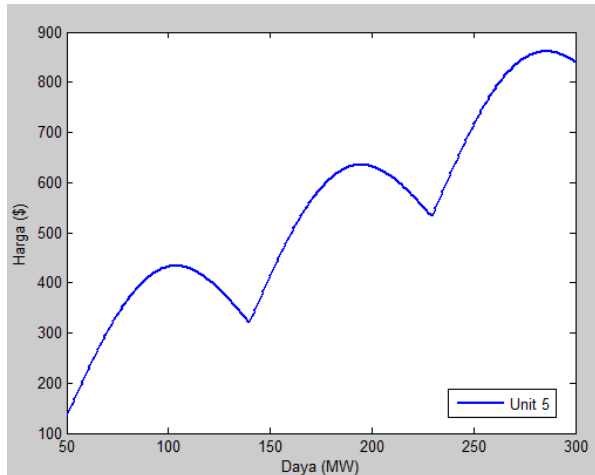
Tabel 4. 5 Hasil Simulasi DED 5 Unit Pembangkit dengan VPE (lanjutan)

13	42.0	106.7	87.1	215.3	252.8	\$1,983.10	704.1
14	29.7	92.4	73.1	194.9	300.0	\$1,969.00	690.1
15	43.6	101.7	108.0	149.3	251.5	\$1,864.90	654.2
16	52.3	111.9	83.0	131.1	201.7	\$1,666.80	580.0
17	31.2	87.1	56.7	152.5	230.7	\$1,676.50	558.0
18	25.7	106.4	84.8	120.9	270.2	\$1,746.90	608.1
19	17.5	116.4	95.3	125.4	299.5	\$1,947.70	654.1
20	16.5	93.8	118.3	175.4	300.0	\$2,203.30	704.2
21	37.5	118.4	148.2	125.8	250.2	\$1,921.20	680.2
22	61.2	121.2	120.9	99.5	202.4	\$1,744.00	605.1
23	46.5	123.3	152.2	50.3	154.8	\$1,572.50	527.1
24	37.9	123.5	155.6	41.1	105.0	\$1,399.90	463.1
Daya						\$42,792.4	14579.5

Tabel 4.5 menunjukkan hasil simulasi DED dengan melibatkan *valve point effect* sehingga kurva pembangkitan tiap generator berbentuk non-linier dengan orde tinggi seperti pada Gambar 4.5 yaitu kurva pembangkitan pada generator 5.

Sistem tenaga listrik dengan 5 Unit pembangkit ini memiliki beban maksimal dan minimal yang mampu disuplai oleh semua pembangkit tanpa melebihi batas oprasi generator. Biaya total pembangkitan dengan pertimbangan *valve point effect* yaitu sebesar **\$ 42,792.40** dan daya yang dibangkitkan setiap jamnya tidak kurang dari permintaan beban tiap jam dalam satu periode dengan total daya terbangkitkan **14579.5 MW**.

Artinya total daya yang terbangkit memenuhi permintaan beban. Sedangkan batasan oprasi dengan mempertimbangkan *ramp rate* seperti terlihat pada **Unit 5 jam ke-14** daya yang dibangkitkan sebesar **300.0 MW**. Pada **jam ke-15** daya yang dibangkitkan sebesar **251.5 MW**. Maka selisih daya terbangkitkan adalah **48.5 MW** dan batasan *ramp rate* untuk **Unit 5** adalah **50** artinya daya yang dibangkitkan setiap jam oleh masing-masing pembangkit masih dalam batasan *ramp rate*. Hal ini menunjukkan bahwa hasil simulasi sudah memenuhi batasan yang digunakan, seperti *equality constrain* ($P_d = P_g$), *inequality constrain* (P_{min} dan P_{max}) serta *ramp-rate* generator.



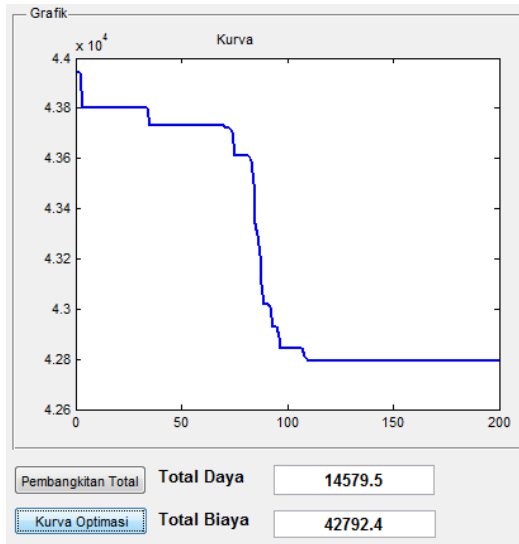
Gambar 4. 5 Kurva Pembangkitan pada Generator 5 dengan 3 Katup Pembukaan

Effek dari *valve point* memberikan perubahan biaya dan jauh lebih mahal yakni \$ **3,060.50** atau **7.7 %** dari total biaya termurah. Maka dari itu *valve point effect* dalam perhitungan DED penting untuk tidak diabaikan. Tabel 4.6 menunjukkan hasil perbandingan antara kasus I dengan kasus II dengan *plant* yang sama.

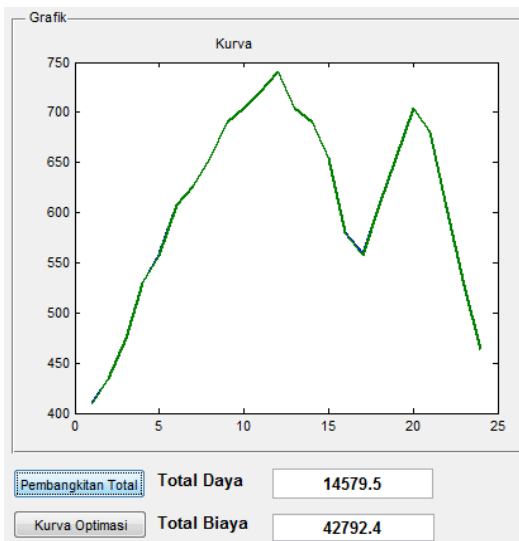
Hasil simulasi DED dengan mempertimbangkan *valve point effect* menggunakan metode ACO untuk mendapatkan biaya minimum pembangkitan ditunjukkan pada Gambar 4.6 kurva konvergensi hasil *running* DED dengan iterasi 200. Gambar 4.6 menunjukkan perhitungan pada iterasi ke 100 sudah menunjukkan konvergensi dengan lama komputasi yakni **470.13 detik**. Gambar 4.7 menunjukkan kurva kestimbangan daya antara pembangkitan dan permintaan beban. Berdasarkan data yang telah disediakan daya yang dibangkitkan lebih besar atau sama dengan daya yang diminta.

Tabel 4. 6 Tabel Perbandingan Pada Plant I

5 Unit Pembangkit	Harga	Waktu Komputasi (s)
Kasus I	\$ 39,731.90	344.87
Kasus II	\$ 42,792,40	470.13



Gambar 4. 6 Konvergensi IEEE 5 Unit Generator dengan *Valve Point Effect*.



Gambar 4. 7 Kurva Kestimbangan Daya Pembangkitan dan Pembebanan

4.3 Sistem Tenaga Listrik 10 Unit Pembangkit

Pada *plant* yang kedua akan dilakukan dua studi kasus simulasi seperti yang sudah dijelaskan sebelumnya.

4.3.1 Simulasi Kasus III Plant 2

Pada kasus III ini akan disimulasikan DED batasan *valve point effect*. Tujuannya agar dapat dibandingkan dengan kasus IV bagaimana pengaruh dari *valve point effect* dan *multiple fuel options* jika dimasukkan kedalam fungsi objektif. Parameter ACO yang digunakan yaitu:

6. Populasi semut (m) sebesar 30
7. Tingkat penguapan feromon (ρ) 0.8
8. Tingkat relatif informasi heuristik (β) sebesar 0.98
9. Jumlah simpul dalam penjabaran graf (n) sebesar 500
10. Intensitas feromon mula-mula (τ) sebesar 0.98

Berikut adalah hasil simulasi DED kasus I yang ditunjukkan pada Tabel 4.7.

Tabel 4. 7 Hasil Simulasi DED (24 jam) Kasus III *Plant* 2 Bagian 1

Jam	Unit						
	P1	P2	P3	P4	P5	P6	P7
1	168.6	230.7	137.3	60.0	73.0	114.2	129.4
2	185.0	209.7	204.9	65.4	74.0	118.8	130.0
3	244.9	253.5	265.4	60.2	73.0	110.0	130.0
4	211.8	319.3	248.2	90.8	115.3	158.7	130.0
5	232.4	386.5	309.7	90.8	98.3	110.0	130.0
6	288.3	446.2	291.8	80.9	127.5	127.3	130.0
7	355.2	455.1	278.1	78.6	103.8	159.4	130.0
8	424.0	458.3	260.2	121.3	108.9	110.4	130.0
9	387.2	396.5	320.0	171.3	158.9	160.0	130.0
10	365.1	437.0	320.0	221.3	208.9	160.0	130.0
11	351.8	435.4	320.0	271.3	243.0	159.5	130.0
12	383.0	449.1	320.0	300.0	243.0	160.0	130.0
13	364.7	456.4	245.6	287.5	233.5	145.8	130.0
14	370.0	459.0	263.7	243.1	187.2	110.0	130.0
15	441.0	421.2	206.9	194.5	137.8	110.0	130.0
16	453.9	349.1	135.9	165.4	87.8	111.0	130.0
17	388.0	406.0	118.2	115.6	90.8	110.0	130.0
18	450.6	451.9	136.0	90.4	89.6	130.5	130.0

Tabel 4. 7 Hasil Simulasi DED (24 jam) Kasus III *Plant 2* Bagian 1 (lanjutan)

Jam	Unit						
	P1	P2	P3	P4	P5	P6	P7
19	455.7	392.2	155.2	134.0	139.6	160.0	130.0
20	460.0	460.0	235.2	184.0	189.6	160.0	130.0
21	458.6	458.6	300.8	135.4	150.6	110.1	130.0
22	389.0	447.8	221.6	87.8	100.6	110.0	130.0
23	318.7	374.0	141.6	63.4	73.0	110.0	130.0
24	304.7	310.4	74.5	60.0	73.0	110.0	130.0

Tabel 4. 7 Hasil Simulasi DED (24 jam) Kasus III *Plant 2* Bagian 2

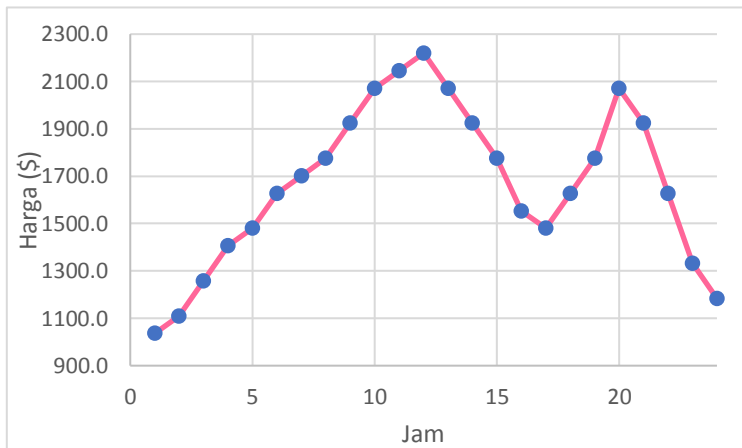
Jam	Unit			Biaya	Daya yang dibangkitkan
	P8	P9	P10		
1	47.1	20.0	55.0	\$28,173.00	1035.1
2	47.1	20.1	55.0	\$29,749.00	1110.0
3	47.0	20.0	55.0	\$32,947.00	1259.0
4	57.0	20.2	55.0	\$36,049.00	1406.3
5	48.3	20.0	55.0	\$37,833.00	1480.9
6	50.2	30.9	55.0	\$40,963.00	1628.0
7	59.2	27.9	55.0	\$42,509.00	1702.2
8	55.1	52.8	55.0	\$44,514.00	1776.0
9	85.1	60.0	55.0	\$47,767.00	1924.1
10	115.1	60.0	55.0	\$51,186.00	2072.4
11	120.0	60.0	55.0	\$52,932.00	2146.0
12	120.0	60.0	55.0	\$54,619.00	2220.1
13	116.9	36.9	55.0	\$51,266.00	2072.2
14	87.0	20.0	55.0	\$47,927.00	1925.0
15	57.0	23.0	55.0	\$44,571.00	1776.4
16	47.0	20.0	55.0	\$39,725.00	1554.9
17	47.1	20.0	55.0	\$37,984.00	1480.6
18	72.8	21.9	55.0	\$41,119.00	1628.7
19	102.8	51.9	55.0	\$44,567.00	1776.4
20	120.0	60.0	55.0	\$50,776.00	2053.8
21	96.0	30.0	55.0	\$47,701.00	1924.9
22	66.0	20.4	55.0	\$41,077.00	1628.1

Tabel 4. 7 Hasil Simulasi DED (24 jam) Kasus III *Plant 2* Bagian 2 (lanjutan)

Jam	Unit			Biaya	Daya yang dibangkitkan
	P8	P9	P10		
23	47.0	20.0	55.0	\$34,626.00	1332.8
24	47.4	20.0	55.0	\$31,478.00	1185.0
Total				\$ 1,012,058.00	40098.9

Simulasi kasus III akan dilakukan sesuai dengan permintaan beban selama 24 jam dengan total beban 40.100 MW yang ditunjukkan pada Gambar 4.8. Pada Tabel 4.7 menunjukkan hasil simulasi DED dengan melibatkan *valve point effect* sehingga kurva pembangkitan tiap generator berbentuk non-linier dengan orde tinggi seperti pada Gambar 4.9 yaitu kurva pembangkitan pada Generator 2. Sistem tenaga listrik dengan 10 Unit pembangkit ini diasumsikan masing-masing 3 tipe bahan bakar yang berbeda, sehingga memberikan 3 jenis fungsi biaya yang berbeda pula [11].

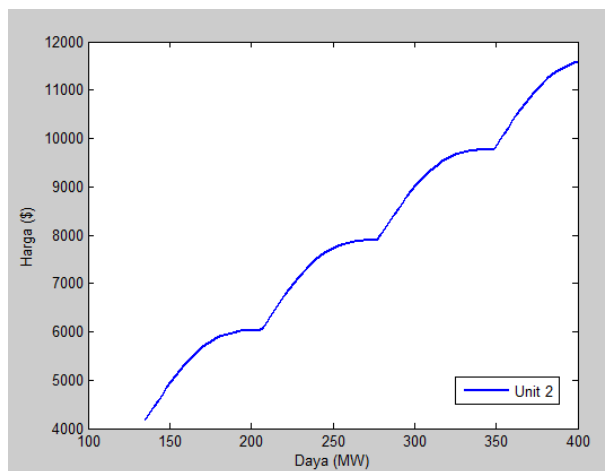
Sistem dengan 10 pembangkit ini memiliki beban maksimal dan minimal yang mampu disuplai oleh pembangkit semua pembangkit tanpa melebihi batas operasi generator dengan biaya total pembangkitan **\$ 1,012,058.00**. Daya yang dibangkitkan setiap jamnya tidak kurang dari permintaan beban tiap jam dalam satu periode yakni **40.098,9 MW**.



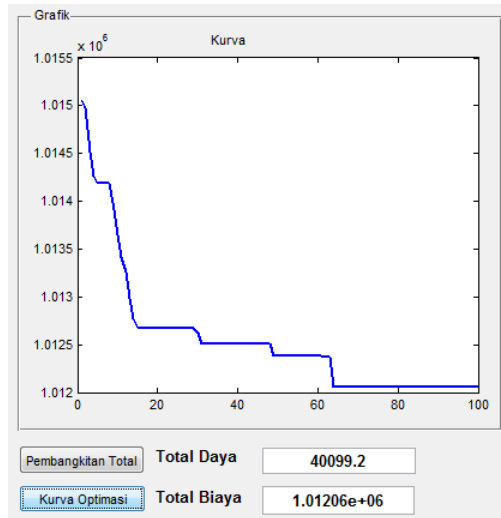
Gambar 4. 8 Kurva Permintaan Beban Satu Periode

Artinya total daya yang terbangkit memenuhi permintaan beban. Sedangkan batasan oprasi dengan mempertimbangkan *ramp rate* seperti terlihat pada **Unit 5 jam ke-15** daya yang dibangkitkan sebesar **137.8 MW**. Pada **jam ke-16** daya yang dibangkitkan sebesar **87.8 MW**. Maka selisih daya terbangkitkan adalah **50 MW** dan batasan *ramp rate* untuk **Unit 4** adalah **50** artinya daya yang dibangkitkan setiap jam oleh masing-masing pembangkit masih dalam batasan *ramp rate*. Artinya total daya yang terbangkit memenuhi permintaan beban. Hal ini menunjukkan bahwa hasil simulasi sudah memenuhi batasan yang digunakan, seperti equality constrain ($P_d = P_g$), inequality constrain (P_{min} dan P_{max}) serta ramp-rate generator.

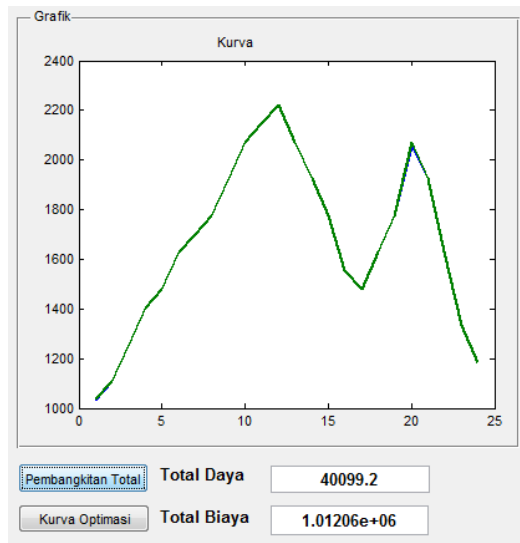
Pada Gambar 4.10 kurva konvergensi hasil *running* DED dengan mempertimbangkan *valve point effect* tanpa mempertimbangkan *multiple fuel options* menggunakan metode ACO untuk iterasi 100. Hal ini menunjukkan hasil dari perhitungan algoritma ACO untuk mendapatkan biaya minimum pembangkitan DED pada iterasi ke 70 sudah menunjukkan konvergensi dengan lama komputasi yakni **2612.82 detik**. Gambar 4.11 menunjukkan kurva kestimbangan daya antara pembangkitan dan permintaan beban dalam satu periode. Berdasarkan data yang telah disediakan daya yang sama dengan daya yang diminta.



Gambar 4. 9 Kurva Pembangkitan pada Generator 2 dengan 4 Katup Pembukaan



Gambar 4. 10 Konvergensi IEEE 10 Unit Generator dengan *Valve Point Effect*.



Gambar 4. 11 Kurva Kestimbangan Daya Pembangkitan dan Pembebanan

4.3.1 Simulasi Kasus IV Plant 2

Pada kasus IV ini akan disimulasikan DED dengan batasan *valve point effect* dan *multiple fuel options*. Kemudian dibandingkan hasil biaya pembangkitan dengan kasus III untuk mengetahui bagaimana pengaruh dari perhitungan DED dengan menggunakan *multiple fuel options* pada fungsi objektif pada yang digunakan. Simulasi kasus IV akan dilakukan pembebanan yang sama pada kasus III selama 24 jam. Parameter ACO sengaja tidak dirubah karena telah menemukan hasil yang optimal serta mempermudah dalam analisa. Berikut adalah hasil simulasi DED kasus IV yang ditunjukkan pada Tabel 4.8.

Tabel 4. 8 Hasil Simulasi DED (24 jam) Kasus IV *Plant 2* Bagian1

Jam	Unit						
	P1	P2	P3	P4	P5	P6	P7
1	155.6	234.0	143.3	60.0	77.4	114.3	129.4
2	161.0	230.0	213.7	60.9	73.5	118.5	130.0
3	182.5	296.2	282.1	62.7	73.1	110.0	130.0
4	190.1	273.3	303.5	98.7	114.7	159.4	130.0
5	258.3	337.6	318.8	76.9	127.4	110.0	130.0
6	194.7	337.1	320.0	126.9	177.4	160.0	130.0
7	269.9	413.6	305.9	93.3	145.2	159.5	130.0
8	202.1	410.8	320.0	143.3	195.2	160.0	130.0
9	184.7	458.2	320.0	193.3	243.0	160.0	130.0
10	264.7	460.0	320.0	243.3	243.0	160.0	130.0
11	306.8	457.9	320.0	293.3	243.0	160.0	130.0
12	380.8	452.9	318.7	299.7	243.0	160.0	130.0
13	411.2	456.7	275.4	265.0	209.8	129.8	130.0
14	413.8	433.1	279.8	242.8	160.2	110.0	130.0
15	429.9	434.9	214.7	193.2	138.6	110.0	130.0
16	436.0	365.7	154.0	143.8	88.6	110.0	130.0
17	373.5	405.9	151.2	94.8	91.3	110.0	130.0
18	416.9	434.5	122.8	64.2	129.8	160.0	130.0
19	434.6	373.3	176.0	114.2	179.8	160.0	130.0
20	453.9	450.4	252.1	161.6	229.2	160.0	130.0
21	446.9	446.4	295.1	138.9	181.5	110.0	130.0
22	391.9	423.0	217.4	88.9	131.5	110.0	130.0
23	314.3	364.6	150.6	60.0	81.5	110.0	130.0
24	275.0	323.1	73.0	73.4	77.6	110.9	130.0

Tabel 4. 8 Hasil Simulasi DED (24 jam) Kasus IV *Plant 2* Bagian 2

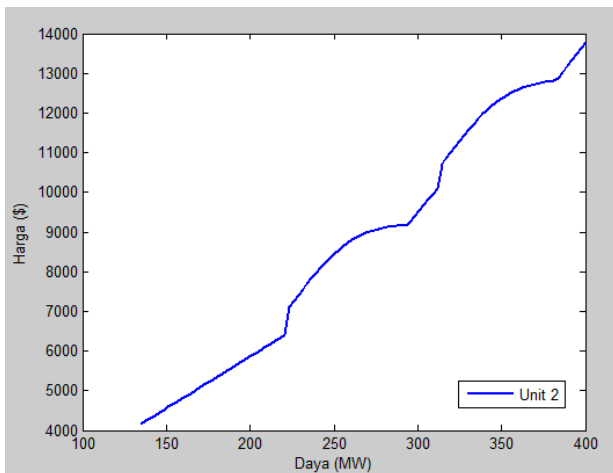
Jam	Unit			Biaya	Daya yang dibangkitkan
	P8	P9	P10		
1	47.0	20.0	55.0	\$28,482.00	1036.0
2	47.0	20.4	55.0	\$30,047.00	1110.0
3	47.1	20.3	55.0	\$33,420.00	1258.9
4	56.4	25.0	55.0	\$36,570.00	1406.0
5	47.0	20.0	55.0	\$38,450.00	1481.0
6	77.0	50.0	55.0	\$41,867.00	1628.0
7	69.6	60.0	55.0	\$43,611.00	1702.0
8	99.6	60.0	55.0	\$45,342.00	1776.1
9	120.0	60.0	55.0	\$48,878.00	1924.2
10	120.0	60.0	55.0	\$51,791.00	2056.0
11	119.9	60.0	55.0	\$53,883.00	2146.0
12	120.0	60.0	55.0	\$55,511.00	2220.1
13	102.9	37.0	55.0	\$51,773.00	2072.8
14	79.9	20.0	55.0	\$48,297.00	1924.5
15	49.9	20.2	55.0	\$44,949.00	1776.4
16	50.8	21.2	55.0	\$39,878.00	1554.9
17	47.3	21.6	55.0	\$38,368.00	1480.6
18	63.2	51.6	55.0	\$41,925.00	1628.0
19	93.2	59.9	55.0	\$45,402.00	1776.0
20	120.0	60.0	55.0	\$52,353.00	2072.3
21	91.2	30.0	55.0	\$48,155.00	1924.9
22	61.2	20.0	55.0	\$41,269.00	1628.9
23	47.0	20.0	55.0	\$34,681.00	1332.9
24	47.0	20.0	55.0	\$31,685.00	1185.0
Total				\$ 1,026,587.00	40101.5

Tabel 4.8 menunjukkan hasil simulasi DED dengan melibatkan *valve point effect* dan *multiple fuel options* sehingga kurva pembangkitan tiap generator berbentuk non-linier dengan orde tinggi seperti pada Gambar 4.12 yaitu kurva pembangkitan pada generator 2. Sistem tenaga listrik dengan 10 Unit pembangkit ini diasumsikan masing-masing 3 tipe bahan bakar yang berbeda, sehingga memberikan 3 jenis fungsi biaya yang berbeda pula. Sistem dengan 10 pembangkit ini memiliki beban maksimal dan minimal yang mampu

disuplai oleh pembangkit semua pembangkit tanpa melebihi batas operasi generator dengan biaya total pembangkitan \$ **1,026,587.00**. Daya yang dibangkitkan setiap jamnya tidak kurang dari permintaan beban tiap jam dalam satu periode yakni **40101.5 MW**.

Artinya total daya yang terbangkit memenuhi permintaan beban. Sedangkan batasan operasi dengan mempertimbangkan *ramp rate* seperti terlihat pada **Unit 2 jam ke-15** daya yang dibangkitkan sebesar **434.9 MW**. Pada **jam ke-16** daya yang dibangkitkan sebesar **365.7 MW**. Maka selisih daya terbangkitkan adalah **69.2 MW** dan batasan *ramp rate* untuk **Unit 2** adalah **80** artinya daya yang dibangkitkan setiap jam oleh masing-masing pembangkit masih dalam batasan *ramp rate*. Artinya total daya yang terbangkit memenuhi permintaan beban. Hal ini menunjukkan bahwa hasil simulasi sudah memenuhi batasan yang digunakan, seperti equality constrain ($P_d = P_g$), inequality constrain (P_{min} dan P_{max}) serta ramp-rate generator. Hal ini menunjukkan bahwa hasil simulasi sudah memenuhi batasan yang digunakan, seperti equality constrain ($P_d = P_g$), inequality constrain (P_{min} dan P_{max}) serta ramp-rate generator.

Effek dari *multiple fuel options* memberikan perubahan biaya dan jauh lebih mahal yakni \$ **14,529.00** atau **1.43 %** dari total biaya tanpa MFO. Maka dari itu *multiple fuel options* dalam perhitungan DED penting untuk tidak diabaikan.

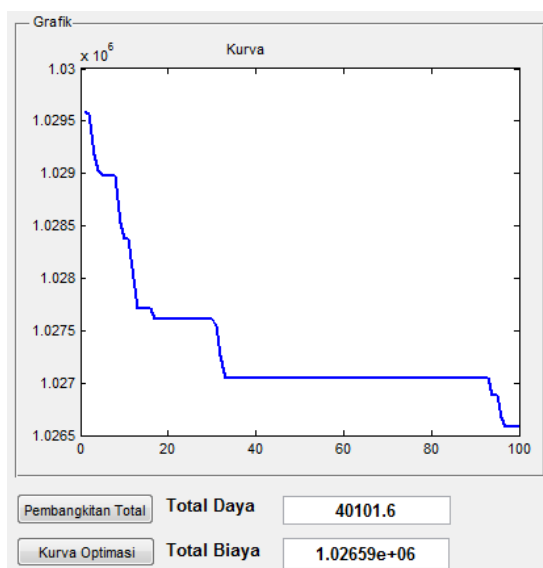


Gambar 4. 12 Kurva Pembangkitan pada Generator 2 dengan pertimbangan VPE dan MFO

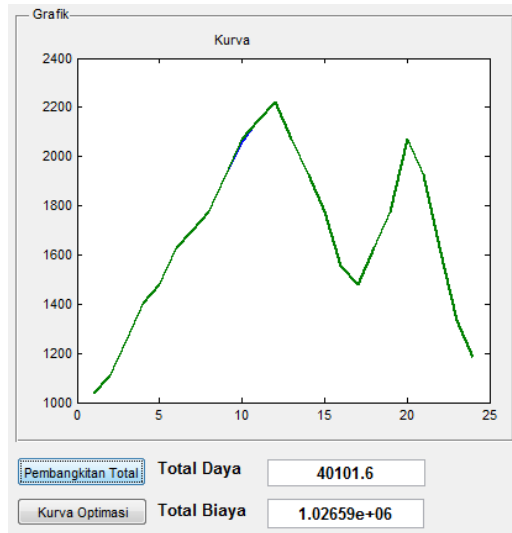
Hasil simulasi DED dengan mempertimbangkan *multiple fuel options* menggunakan metode ACO untuk mendapatkan biaya minimum pembangkitan ditunjukkan pada Gambar 4.13 kurva konvergensi hasil *running* DED dengan iterasi 150. Gambar 4.13 menunjukkan perhitungan pada iterasi ke 100 sudah menunjukkan konvergensinya dengan lama komputasi yakni **2612.82 detik**. Gambar 4.14 menunjukkan kurva kestimbangan daya antara pembangkitan dan permintaan beban. Berdasarkan data yang telah disediakan daya yang dibangkitkan lebih besar atau sama dengan daya yang diminta.

Tabel 4. 9 Perbandingan Pada Plant II

10 Unit Pembangkit	Harga	Waktu Komputasi (s)
Kasus III	\$ 1,012,058.00	2554.83
Kasus IV	\$ 1,026,587.00	2612.82



Gambar 4. 13 Konvergensi IEEE 10 Unit Generator dengan VPE dan MFO



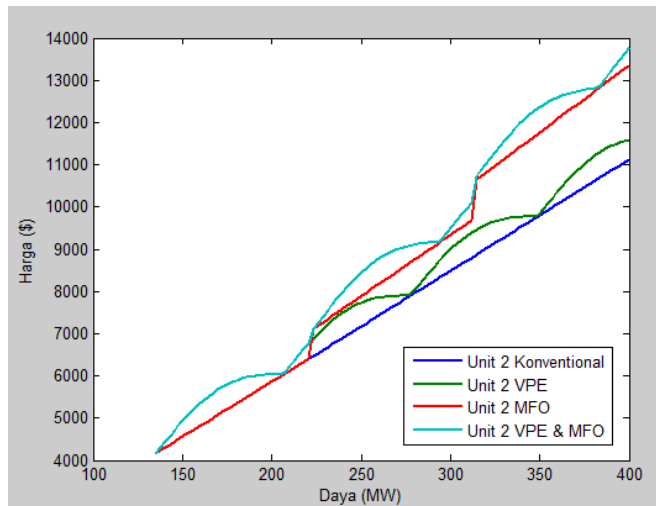
Gambar 4. 14 Kurva Kestimbangan Daya Pembangkitan dan Pembebanan

Pengaruh dari penambahan parameter yaitu VPE dan MFO menyebabkan kurva biaya pembangkitan yang tidak mulus. Hal ini dapat dilihat pada Gambar 4.15 yaitu perbandingan tanpa penambahan dan penambahan parameter VPE dan MFO pada Unit 2 untuk Plant II sistem dengan 10 Unit Generator. Sehingga menyebabkan biaya pembangkitan lebih mahal dengan adanya parameter VPE dan MFO. Penambahan parameter yang dilakukan bertujuan untuk mendapatkan perhitungan optimasi dengan ketepatan yang sebaik mungkin berdasarkan dengan kondisi pembangkit yang sebenarnya.

Tabel 4. 10 Komparasi Metode untuk 10 Unit Pembangkit

Metode	Total Biaya (\$)	Komsumsi (Min)
SQP	\$1,051,163.00	1.19
EP	\$1,048,638.00	42.29
Hybrid	\$1,035,748.00	20.51
MIQP	\$1,027,768.00	4.5
ACO	\$1,026,587.00	43.53

Metode ACO mampu bekerja secara optimal untuk sistem IEEE 5 Unit Generator seperti yang terlihat pada Tabel 4.10 sebagai komparasi dengan metode pembandingan lainnya yaitu *Mixed Integer Quadratic Program* (MIQP) serta metode lainnya [11]. Hasil yang didapatkan dengan menggunakan metode ACO biaya pembangkitan sudah mendapatkan hasil yang optimal untuk sistem IEEE 10 Unit Pembangkit yakni jauh lebih murah namun waktu komputasinya relative lebih lama.



Gambar 4. 15 Perbandingan Parameter Kurva Pembangkitan pada Generator 2

Halaman ini sengaja dikosongkan

BAB 5

PENUTUP

5.1. Kesimpulan

Dari pengujian permasalahan *dynamic economic dispatch*, (DED) menggunakan metode *ant colony optimization*, (ACO) dapat diambil kesimpulan sebagai berikut:

1. Parameter merupakan elemen yang sangat penting dalam implementasi algoritma ACO. Penentuan parameter ACO berdasarkan banyaknya jumlah pembangkit serta besarnya daya maksimum dan minimum yang dibangkitkan oleh masing-masing generator. Kombinasi dari parameter ACO yang tepat mempengaruhi efektifitas dalam hal lain seperti waktu komputasi serta optimasi dari fungsi objektif yang dihasilkan (konvergen).
2. Dilihat dari segi waktu komputasi, algoritma yang menggunakan ACO ini masih sangat tidak efisien dibanding metode lain. Hal ini disebabkan algoritma yang dibangun pada tugas akhir ini sangat banyak mengandung perulangan. Sehingga memberikan waktu komputasi lebih lama. Semakin besar selisih daya yang dibangkitkan generator ($P_{\min} - P_{\max}$) maka simpul yang dibangun juga harus lebih besar seperti pada *plant I* dan *plant II*.
3. Pengaruh dari *valve-point effect*, (VPE) yang menghasilkan kurva pembangkitan *non-linear* dengan orde tinggi menyebabkan biaya pembangkitan menjadi lebih besar seperti yang telah disimulasikan pada sistem IEEE 5 Unit pembangkit untuk kasus I dan kasus II. Fungsi biaya dengan penambahan VPE memberikan selisih 7.7 % atau sebesar \$ 3,060.50 dari total harga perhitungan DED tanpa mempertimbangkan VPE. Maka kendala ini penting untuk tidak diabaikan dalam perhitungan optimasi DED.
4. Pengaruh dari *multiple fuel options*, (MFO) yang menghasilkan kurva pembangkitan *piecewise* berdasarkan jenis bahan bakar yang digunakan menyebabkan biaya pembangkitan menjadi lebih besar seperti yang telah disimulasikan pada sistem IEEE 10 Unit pembangkit untuk kasus III dan kasus IV. Fungsi biaya dengan penambahan MFO memberikan selisih 1.43 % atau sebesar \$ 14,529.00 dari total harga perhitungan DED dengan hanya mempertimbangkan VPE. Maka kendala ini penting untuk tidak diabaikan dalam perhitungan optimasi DED

5.2. Saran

1. Saran untuk kedepannya algoritma ini bisa dimodifikasi sehingga bisa lebih efisien untuk waktu komputasinya.
2. Mengimplementasikan algoritma ACO pada permasalahan sistem nyata.

DAFTAR PUSTAKA

- [1] L. Mohammad Ali, "Modification of multi-area economic dispatch with multiple fuel options, considering the fuelling limitations", IET Gener. Transm. Distrib., vol. 8, pp. 1098–1106, December 2013.
- [2] Lazuardi. Dika, "Dynamic Economic Dispatch Dengan Mempertimbangkan Kerugian Transmisi Menggunakan Metode Sequential Quadratic Program", ITS Surabaya, 2016.
- [3] Mahatmya. Atya, "Implementasi Algoritma Ant Colony Optimization Untuk Menyelesaikan Permasalahan Dynamic Economic Dispatch Dengan Memperhatikan Rugi Rugi Daya Transmisi dan Valve Point Effect", ITS Surabaya, vol. 1, no. 1, pp. 1–6, 2012.
- [4] Hadi Saadat, *Power System Analysis*, Third Edition.
- [5] Raul E. Perez-Guerrero and Jose R. Cedenio, "Economic Power Dispatch with Non-Smooth Cost Functions Using Differential Evolution", Member IEEE., pp. 191 – 197, 2012.
- [6] R. Balamurugan, "Self-Adaptive Differential Evolution Based Power Economic Dispatch of Generators with Valve-Point Effect and Multiple Fuel Options", International Conference on Emerging Trends in Electrical Engineering and Energy Management (ICETEEEM)., pp. 191 – 197, 2012.
- [7] M. Dorigo, M. Vincent, and A. Colomi, "The ant system optimization by a colony of cooperating agents", IEEE Transactions on Systems, Man & Cybernetics B, 1996, vol. 26, NO. 2, pp. 29-41.
- [8] S. Vijayaraj, Dr. R. K. Santhi, "Multi-Area Economic Dispatch with Multi Fuel Option Using Improved Bat Algorithm", IJIRS, Engineering and Technology, vol. 5, pp. 2347–7610, October 2007.
- [9] Saravuth. Pothiya, Issarachai. Ngamroo, "Ant colony optimisation for economic dispatch problem with non-smooth cost functions", IEE Trans. Power sys, vol. 17, No. 2, May 2002

- [10] F.S. Abu-Mouti, M.E. El-Hawary, "Optimal Dynamic Economic Dispatch Including Renewable Energy Source Using Artificial Bee Colony Algorithm", Systems Conference (SysCon), 2012 IEEE International, pp.1,6, 19-22 March 2012
- [11] M. Q. Wang, X. S. Han, M. Yang and M. X. Wang, "Dynamic Economic Dispatch with Valve Point Effect, Prohibited Operation Zones, and Multiple Fuel Option", IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)., pp. 1-5, Feb. 2014.
- [12] Hemamalini. S, "Dynamic Economic Dispatch with Valve Point Effect Using MacLaurin Series Based Lagrangian Method", National of Institute Technology, TamilNadu India, IJCA vol. 1-17, 2010.

LAMPIRAN

I. Tampilan GUI input Matlab

Tampilan input data Plant 5 Generator

Dynamic Economic Dispatch Using Ant Colony Optimization
Mochamad Luthfiyansah
2215 105 063

Variabel Kontrol Algoritma

Iterasi: 200 r0: 0.9
Jumlah Semut: 30 Beta: 0.7

Button Group

Konvensional Valve Point Effect Multiple Fuel Options
[Dispatch] [Dispatch] [Dispatch]

Karakteristik Unit Input Output

A (\$/MW ²)	B (\$/MW)	C (MW)	e	f	Batas Bawah	Batas
0.0080	2	25	100	0.0420	10	
0.0030	1.8000	60	140	0.0400	20	
0.0012	2.1000	100	160	0.0380	30	
1.0000e-03	2	120	180	0.0370	40	
0.0015	1.8000	40	200	0.3500	50	

Tambah Generator Kurangi Generator Data 10 unit Data 5 Unit

Daya Permintaan

Jam	Daya
Jam 1	410
Jam 2	435
Jam 3	475
Jam 4	530
Jam 5	558
Jam 6	608
Jam 7	626
Jam 8	654

Tampilan input data Plant 10 Generator

Dynamic Economic Dispatch Using Ant Colony Optimization
Mochamad Luthfiyansah
2215 105 063

Variabel Kontrol Algoritma

Iterasi: 100 r0: 0.9
Jumlah Semut: 30 Beta: 0.7

Button Group

Konvensional Valve Point Effect Multiple Fuel Options
[Dispatch] [Dispatch] [Dispatch]

Karakteristik Unit Input Output

A (\$/MW ²)	B (\$/MW)	C (MW)	e	f	Batas Bawah	B
4.3000e-04	21.6000	958.2000	450	0.0410	150	
6.3000e-04	21.0500	1.3136e+03	600	0.0360	135	
3.9000e-04	20.8100	604.9700	320	0.0280	73	
7.0000e-04	23.9000	471.6000	260	0.0520	60	
7.9000e-04	21.6200	480.2900	280	0.0630	73	
0.0048	23.2300	639.4000	340	0.0820	47	

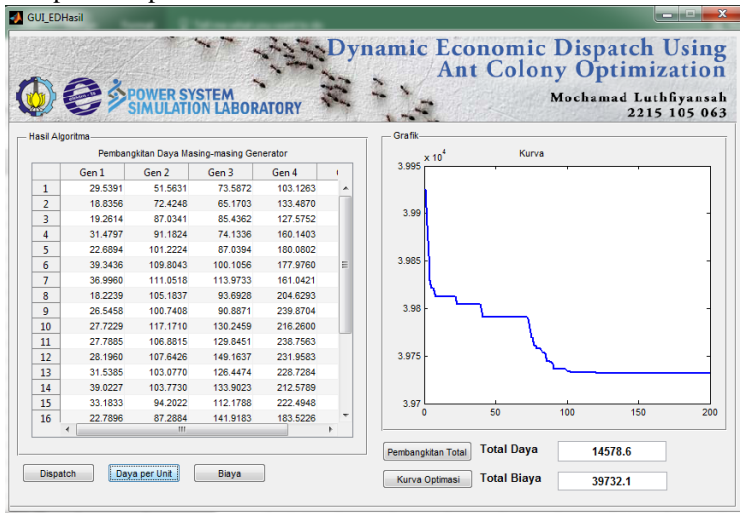
Tambah Generator Kurangi Generator Data 10 unit Data 5 Unit

Daya Permintaan

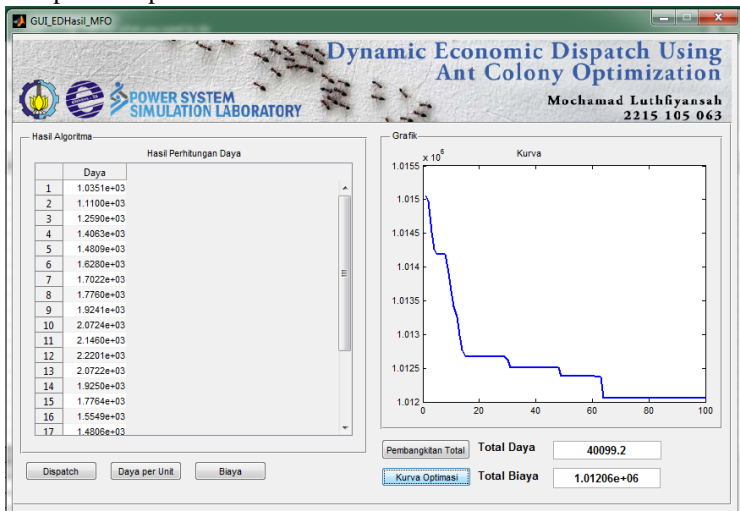
Jam	Daya
Jam 1	1036
Jam 2	1110
Jam 3	1258
Jam 4	1406
Jam 5	1480
Jam 6	1628
Jam 7	1702
Jam 8	1776

II. Tampilan GUI output Matlab

Tampilan output data Plant 5 Generator



Tampilan output data Plant 10 Generator



III. M-file Ant Colony Optimization

```

% M. LUTHFIYANSAH
% ANT COLONY OPTIMIZATION
% ECONOMIC DISPATCH

function[sumpemb, pemb, rekamharga,
memoribest]=Cinta(sbrdata, d, ite, ant, ro,
beta)

clc
% clear all
tic;

%program konstruksi graf
%inisialisasi konstruksi

% ite
% ant

%% Input Data
n=500; %banyak node per generator

harga=sbrdata(:,1:5); %input data tabel koef
a,b,c
gen=sbrdata(:,6:7); %input data tabel min max
ramprate=sbrdata(:,8:9)'; %input data ramp-
rate

%%
w=size(d,1); %jumlah beban
g=size(gen,1); %jumlah generator

%% Inisialisasi Ant Colony

% ite=150; % banyak iterasi -control
% ant=30; % banyak semut -control
% ro=0.2; % -control
q0=0.5;

tau0=0;

```

```

alpha=1;
% beta=0.9; % -control
total=[];

%inisialisasi total
total(1:ite,1:ant)=0;

pemb=[];
hargaperjam=[];
hargasbl=0;

%% beban ke 1
for z=1:w
    %% inisialisasi
    fprintf('Menghitung pembangkit jam ke %d\n',z);

    sumnod=[];
    par=[];
    pilih=[];
    mini=[];
    maxi=[];
    mem=[];
    mem=[];
    prob=[];
    sumnod=[];
    titik=[];
    cost=[];
    pilih=[];
    faktor=1; %safety while
function
    vpe=[];
    vpe2=[];
    tau=[];
    tau(1:g,1:n)=0.8; %inisialisasi nilai
tau

    %% Batas Generator dan Ramp-Rate
    batasgen=[];
    if z>1

```

```

        for x=1:g
%generator ke-
            y=0;
            y=pemb(z-1,x)-ramprate(2,x);
%pemb pada jm 1 utk gen 1-gen
            if y<gen(x,1)
%jika pemb < max gen
                batasgen(x,1)=gen(x,1);
%batas gen max=
            else
                batasgen(x,1)=y;
            end
            y=0;
            y=pemb(z-1,x)+ramprate(1,x);
            if y>gen(x,2)
                batasgen(x,2)=gen(x,2);
            else
                batasgen(x,2)=y;
            end
        end
    else
        batasgen=gen;
    end
%    batasgen=[batasgen;0 0];

%% Membangkitkan level daya dari tiap
generator

    for i=1:g
        for j=1:n
            titik(i,j)=batasgen(i,1)+(j-
1)*( (batasgen(i,2)-batasgen(i,1))/(n-1));
        end

    end
% masukan nilai biaya

    for i=1:g

        batasgen1=batasgen(i,1)+(0.333*(batasgen(i,2)-
        batasgen(i,1)));
    end
end

```

```

batasgen2=batasgen(i,1)+(0.667*(batasgen(i,2)-
batasgen(i,1)));
    for j=1:n
        % conventional
        % cost(i,j)=
        harga(i,1)*titik(i,j)^2+harga(i,2)*titik(i,j)+
        harga(i,3);

        %VPE

vpe2=sin(harga(i,5)*(batasgen(i,1)-
titik(i,j)));
        vpe=abs(harga(i,4)*vpe2);
        cost(i,j)=
        harga(i,1)*titik(i,j)^2+harga(i,2)*titik(i,j)+
        harga(i,3)+vpe;

%           %MFO & VPE
%           % i = generator, j = step
kenaikan daya cek minim dulu
%           if titik(i,j) >= batasgen(i,1)
&& titik(i,j) <= batasgen1
%               cost(i,j)=
        harga(i,1)*titik(i,j)^2+harga(i,2)*titik(i,j)+
        harga(i,3)+abs(harga(i,4)*sin(harga(i,5)*(bata
sge(i,1)-titik(i,j))));
%           elseif titik(i,j) >= batasgen1
&& titik(i,j) <= batasgen2
%               cost(i,j)=
        (1.1*harga(i,1))*titik(i,j)^2+(1.1*harga(i,2))
        *titik(i,j)+(1.1*harga(i,3))+abs(harga(i,4)*si
n(harga(i,5)*(batasgen1-titik(i,j))));
%           else
%               cost(i,j)=
        (1.2*harga(i,1))*titik(i,j)^2+(1.2*harga(i,2))
        *titik(i,j)+(1.2*harga(i,3))+abs(harga(i,4)*si
n(harga(i,5)*(batasgen2-titik(i,j))));
%           end
    end
end

```

```

%% iterasi ant colony
    for t=1:ite
        p=[];
        pilih=[];
        for m=1:ant
            %% state transition rule
            q=unifrnd(0,1); %nilai random
            q
            % q=0.4;
            if q <= q0 %semut
                melakukan eksploitasi versi satu
                    for i=1:g-1
                        for j=1:n
                            par(i,j)=
                                tau(i,j)^alpha*((1/cost(i,j))^beta); %simpul
                                yang akan dipilih semut
                        end
                        [a,b]= max(par(i,:));
                    %pemilihan jalur dimulai dari nilai simpul
                    %tertinggi(feromon terbesar)
                    mem(m,i)=b;

                %memori simpul

                pilih(m,i)=titik(i,mem(m,i)); %simpul
                terbesar=harga terrenndah

                sumnod(t,m)=sum(pilih(m,:));

                minimum=sum(batasgen(i+1:g,1));

                maximum=sum(batasgen(i+1:g,2));
                mini=(d(z,1)-sumnod(t,m))-
                minimum;
                maxi=(d(z,1)-sumnod(t,m))-
                maximum;

                while (mini<0 || maxi>0)
                    && faktor < n %aturan MINI MAXI
                        par(i,mem(m,i))= -1;
                    %simpul yang telah dilewati oleh semut
                        [a,b]=max(par(i,:));
                        mem(m,i)=b;

```



```

pilih(m,i)=titik(i,mem(m,i));

sumnod(t,m)=sum(pilih(m,:));
mini=(d(z,1)-
sumnod(t,m))-minimum;
maxi=(d(z,1)-
sumnod(t,m))-maximum;
faktor=faktor+1;
end
faktor=1;
total(t,m) = total(t,m) +
cost(i,b); %total harga pembangkitan
yang diperoleh tiap semut
tau(i,b)=(1-
ro)*tau(i,b)+ro*tau0; %update feromon

end

else %semut
melakukan eksplorasi versi dua
for i=1:g-1
for k=1:n
par(i,k)=
tau(i,k)^alpha*((1/cost(i,k))^beta);
end
ktj=sum(par(i,:));
k=0;
for j=1:n
k=k+1;

prob(i,j)=par(i,k)/ktj; %simoul yang
akan dipilih semut
end
[a,b]=max(prob(i,:));
%pemilihan jalur dimulai dari nilai simpul
tertinggi(feromon terbesar)
mem(m,i)=b;

%memori simpul

pilih(m,i)=titik(i,mem(m,i)); %simput
terbesar=harga terrenndah

```

```

sumnod(t,m)=sum(pilih(m,:));

minimum=sum(batasgen(i+1:g,1));

maximum=sum(batasgen(i+1:g,2));
mini=(d(z,1)-sumnod(t,m))-
minimum;
maxi=(d(z,1)-sumnod(t,m))-
maximum;
while (mini<0 || maxi>0)
&& faktor < n %aturan MINI MAXI
    prob(i,mem(m,i))= -1;
%simpul yang telah dilewati oleh semut
    [a,b]=max(prob(i,:));
    mem(m,i)=b;

pilih(m,i)=titik(i,mem(m,i));

sumnod(t,m)=sum(pilih(m,:));
mini=(d(z,1)-
sumnod(t,m))-minimum;
maxi=(d(z,1)-
sumnod(t,m))-maximum;
    faktor=faktor+1;
end

    faktor=1;
    mem(m,i)=b;
    total(t,m) = total(t,m) +
cost(i,b); %total harga pembangkitan yang
diperoleh tiap semut
    tau(i,b)=(1-
ro)*tau(i,b)+ro*tau0; %update feromon
end
end

%% akhir dari eksplorasi
p(m)=d(z,1)-sumnod(t,m);
nilaikurang=p(m);
i=1;

```

```

        while ((titik(g,i)<nilaikurang) &&
(i<n))
            i=i+1;
        end
        nodl=i;
        mem(m,g)=nodl;

total(t,m)=total(t,m)+cost(g,nodl);
        tau(g,nodl)=(1-
ro)*tau(g,nodl)+ro*tau0;
        if t==1
            [tot,bes]=min(total(t,1:m));
            bestcost=tot;
            best=mem(bes,:);
        else
            [tot,bes]=min(total(t,1:m));
            if tot < bestcost
                bestcost=tot;
                best=mem(bes,:);
            end
        end
        end

        %% memori best berubah sesuai
iterasi

        memoribest(t)=bestcost;
        pathbest(t,:)=best;

%update feromon global
        for i=1:g
            for j=1:n
                if j==best(i)

tau(i,j)=ro*tau(i,j)+(1/bestcost);
                    end
                end
            end
        end

        %% akhir dari pembangkitan ant
population

```

```

        if t==1
            semua=mem;
        else
            semua=[sema;mem];
        end
        for nnn=1:g

            daya(1,nnn)=titik(nnn,best(1,nnn));
            end
            if t==1
                [tot,bes]=min(total(t,1:m));
                bestcost=tot;
                best=mem(bes,:);
            else
                [tot,bes]=min(total(t,1:m));
                if tot < bestcost
                    bestcost=tot;
                    best=mem(bes,:);
                end
            end

            memoribest(t)=bestcost;
            pathbest(t,:)=best;

            %update feromon global
            for i=1:g
                for j=1:n
                    if j==best(i)

                        tau(i,j)=ro*tau(i,j)+(1/bestcost);
                    end
                end
            end

        end
        %% akhir iterasi ant colony

        if t==1
            semua=mem;
        else
            semua=[sema;mem];

```

```

end

pemb=[pemb; daya];
rekamharga(z,1)=bestcost-hargasbl;
hargasbl=bestcost;

end
toc;
%% akhir beban ke-

sumpemb=sum(pemb,2);
disp(pemb);
disp(bestcost);

IV. M-file GUI input Matlab
function varargout = GUI_EDinput(varargin)
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',
mfilename, ...
                  'gui_Singleton',
gui_Singleton, ...
                  'gui_OpeningFcn',
@GUI_EDinput_OpeningFcn, ...
                  'gui_OutputFcn',
@GUI_EDinput_OutputFcn, ...
                  'gui_LayoutFcn',    [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback =
str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] =
gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

```

```

% --- Executes just before GUI_EDinput is made
visible.
function GUI_EDinput_OpeningFcn(hObject,
eventdata, handles, varargin)
% Choose default command line output for
GUI_EDinput
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

clc
% Tampilan header Header -----
-----
head=imread('headerFIX.png');
imshow(head,'Parent', handles.axesHEAD);
axis(handles.axesHEAD,'off');

% Inisialisasi Data Awal -----
-----
set(handles.edit1,'String',0);
%Iterasi
set(handles.edit2,'String',0);
%Jumlah semut
%
set(handles.uitable1,'Data', zeros(3,9));
%Data generator
set(handles.uitable2,'Data', zeros(1,1));
%Daya permintaan / demand
%
set(handles.edit3,'String',0);
%ro
set(handles.edit8,'String',0);
%beta
uiwait(handles.figure1);

function varargout =
GUI_EDinput_OutputFcn(hObject, eventdata,
handles)

```

```

varargout{1} = get(handles.edit1, 'String');
% Iterasi
varargout{2} = get(handles.edit2, 'String');
% Jumlah Semut
varargout{3} = get(handles.edit3, 'String');
% ro
varargout{4} = get(handles.edit8, 'String');
% beta
varargout{5} = get(handles.uitable1, 'Data');
% data pembangkit
varargout{6} = get(handles.uitable2, 'Data');
% kebutuhan daya
varargout{7} = get(handles.figure1, 'Visible');
% tampilan menu utama
delete(hObject);

```

```

function pushbuttonNON_Callback(hObject,
 eventdata, handles)
set(handles.figure1, 'Visible', 'off');
GUI_EDHasil('GUIInput', handles.figure1);
set(handles.figure1, 'Visible', 'on');

```

```

function pushbuttonVPE_Callback(hObject,
 eventdata, handles)
set(handles.figure1, 'Visible', 'off');
GUI_EDHasil_VPE('GUIInput', handles.figure1);
set(handles.figure1, 'Visible', 'on')

```

```

function pushbuttonMFO_Callback(hObject,
 eventdata, handles)
set(handles.figure1, 'Visible', 'off');
GUI_EDHasil_MFO('GUIInput', handles.figure1);
set(handles.figure1, 'Visible', 'on')

```

```

function pushbutton3_Callback(hObject,
 eventdata, handles)
d = get(handles.uitable1, 'Data');
[a b]=size(d);
d = [d; zeros(1,b)];
set(handles.uitable1, 'Data', d);
set(handles.pushbutton5, 'Enable', 'on');

```

```

function pushbutton5_Callback(hObject,
 eventdata, handles)
d = get(handles.uitable1, 'Data');
[a b]=size(d);
if a>1
    c = d(1:a-1,:);
    set(handles.uitable1, 'Data', c);
    if a==2
        set(handles.pushbutton5, 'Enable',
'off');
    end;
end;

function pushbuttonLOAD_Callback(hObject,
 eventdata, handles)
data=[
    0.008    2.0 25   100 0.042    10   75   30   30
    0.003    1.8 60   140 0.04     20  125   30   30
    0.0012   2.1 100  160 0.038    30  175   40   40
    0.001    2.0 120  180 0.037    40  250   50   50
    0.0015   1.8 40   200 0.035    50  300   50   50
];

Pd=[410 435 475 530 558 608 626 654 690 704
720 740 704 690 654 580 558 608 654 704 680
605 527 463]; %/*beban permintaan daya./
set(handles.uitable1, 'Data', data);
set(handles.uitable2, 'Data', Pd);
set(handles.edit1, 'String', 200);
%Iterasi
set(handles.edit2, 'String', 30);
%Jumlah semut
%
set(handles.edit3, 'String', 0.9);
%ro
set(handles.edit8, 'String', 0.7);
%beta

```



```

function pushbuttonLOAD3_Callback(hObject,
eventdata, handles)
data=[
    0.00043    21.6    958.2    450 0.041
150 460 80    80 %1
    0.00063    21.05    1313.6    600 0.036
135 460 80    80 %2
    0.00039    20.81    604.97    320 0.028
73  320 80    80 %3
    0.0007     23.9     471.6     260 0.052
60  300 50    50 %4
    0.00079    21.62    480.29    280 0.063
73  243 50    50 %5
    0.0048     23.23    639.4     340 0.082
47  120 30    30 %8
    0.00056    17.87    601.75    310 0.048
110 160 50    50 %6
    0.10908    19.58    455.6     270 0.098
20  60  30    30 %9
    0.00211    16.51    502.7     300 0.086
129 130 30    30 %7
    0.00951    22.54    692.4     380 0.094
55  55  30    30 %10
];
% Pd=[1036]; %/*beban permintaan daya./
Pd=[1036 1110 1258 1406 1480 1628 1702 1776
1924 2072 2146 2220 2072 1924 1776 1554 1480
1628 1776 2072 1924 1628 1332 1184]; %/*beban
permintaan daya./
set(handles.uitable1, 'Data', data);
set(handles.uitable2, 'Data', Pd);
set(handles.edit1, 'String', 200);
%Iterasi
set(handles.edit2, 'String', 30);
%Jumlah semut
%
set(handles.edit3, 'String', 0.9);
%ro
set(handles.edit8, 'String', 0.7);
%beta

```

```

function edit5_CreateFcn(hObject, eventdata,
handles)
if ispc &&
isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function edit4_Callback(hObject, eventdata,
handles)

function edit4_CreateFcn(hObject, eventdata,
handles)
if ispc &&
isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function edit3_Callback(hObject, eventdata,
handles)

function edit3_CreateFcn(hObject, eventdata,
handles)
if ispc &&
isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function edit7_Callback(hObject, eventdata,
handles)

function edit7_CreateFcn(hObject, eventdata,
handles)
if ispc &&
isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

```

```

function edit8_Callback(hObject, eventdata,
handles)

function edit8_CreateFcn(hObject, eventdata,
handles)
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit1_Callback(hObject, eventdata,
handles)

function edit1_CreateFcn(hObject, eventdata,
handles)
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata,
handles)

function edit2_CreateFcn(hObject, eventdata,
handles)
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function figure1_CloseRequestFcn(hObject,
eventdata, handles)
if isequal(get(hObject, 'waitstatus'),
'waiting')
    uiresume(hObject);
else
    delete(hObject);
end

```

```
function figure1_CreateFcn(hObject, eventdata, handles)
```

```
function uitable1_CreateFcn(hObject, eventdata, handles)
```

V. M-file GUI output Matlab

```
gui_Singleton = 1;
gui_State = struct('gui_Name',
    mfilename, ...
                   'gui_Singleton',
    gui_Singleton, ...
                   'gui_OpeningFcn',
    @GUI_EDHasil_OpeningFcn, ...
                   'gui_OutputFcn',
    @GUI_EDHasil_OutputFcn, ...
                   'gui_LayoutFcn', [] , ...
                   'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] =
gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

function GUI_EDHasil_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
```

```

% Pesan Tunggu -----
-----
handleToMessageBox = msgbox('Sedang dilakukan
perhitungan.','Silakan Tunggu', 'warn');

awal=find(strcmp(varargin, 'GUIInput'));
handles.GUIInput123=varargin{awal+1};
mainHandles=guidata(handles.GUIInput123);

iterasi=get(mainHandles.edit1,'String');
semut=get(mainHandles.edit2,'String');
ro=get(mainHandles.edit3,'String');
beta=get(mainHandles.edit8,'String');

iterasi=str2num(iterasi);
semut=str2num(semut);
ro=str2num(ro);
beta=str2num(beta);

datainput=get(mainHandles.uitable1,'Data');
butuhdaya=get(mainHandles.uitable2,'Data');

[dayajam, dayaunit,biayajam,
optimasi]=Harapan(datainput, butuhdaya,
iterasi, semut, ro, beta);

% Akhir Pesan Tunggu -----
-----
if exist('handleToMessageBox', 'var')
    delete(handleToMessageBox);
    clear('handleToMessageBox');
end

set(handles.pushbutton3,'UserData', dayajam);
set(handles.pushbutton4,'UserData', dayaunit);
set(handles.pushbutton6,'UserData', biayajam);

```

```

set(handles.pushbutton2,'UserData', optimasi);

set(handles.edit2,'String', sum(dayajam));
set(handles.edit1,'String', sum(biayaajam));

set(handles.judultabel,'String','Hasil
Perhitungan Daya');
set(handles.uitable1, 'Data', dayajam);
namakolom=['Daya'];
set(handles.uitable1, 'ColumnName',
namakolom);

set(handles.judulgrafik,'String','Kurva
Optimasi');
plot(handles.axes2, optimasi);
grid on;
guidata(hObject, handles);
uiwait(hObject);

function varargout =
GUI_EDHasil_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;
delete(hObject);

function uitable1_CreateFcn(hObject,
eventdata, handles)

function pushbutton3_Callback(hObject,
eventdata, handles)
set(handles.judultabel,'String','Hasil
Perhitungan Daya');
nilai3=get(handles.pushbutton3,'UserData');
set(handles.uitable1, 'Data', nilai3);
namakolom=['Daya'];
set(handles.uitable1, 'ColumnName',
namakolom);

```

```

function pushbutton4_Callback(hObject,
eventdata, handles)
set(handles.judultabel, 'String', 'Pembangkitan
Daya Masing-masing Generator');
nilai4=get(handles.pushbutton4, 'UserData');
set(handles.uitable1, 'Data', nilai4);

namakolom=[];
a=size(nilai4,2);
for n=1:a
    b=['Gen ', num2str(n)];
    namakolom{n,1}=b;
end
% namakolom=['Gen 1'; 'Gen 2'; 'Gen 3'; 'Gen 4';
'Gen 5'];
set(handles.uitable1, 'ColumnName',
namakolom);

function pushbutton6_Callback(hObject,
eventdata, handles)
set(handles.judultabel, 'String', 'Biaya
Pembangkitan');

nilai6=get(handles.pushbutton6, 'UserData');
set(handles.uitable1, 'Data', nilai6);

namakolom=['Biaya'];
set(handles.uitable1, 'ColumnName',
namakolom);

function edit2_Callback(hObject, eventdata,
handles)

function edit2_CreateFcn(hObject, eventdata,
handles)
if ispc &&
isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

```

```

function edit1_Callback(hObject, eventdata,
handles)

function edit1_CreateFcn(hObject, eventdata,
handles)
if ispc &&
isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function pushbutton1_Callback(hObject,
eventdata, handles)
mainHandles=guidata(handles.GUIInput123);

hasil=get(handles.pushbutton3, 'UserData');
permintaan=get(mainHandles.uitable2, 'Data');

pembangkitan=[hasil'; (permintaan(:,1))'];
pembangkitan=pembangkitan';
plot(handles.axes2,
pembangkitan, 'LineWidth', 2);
set(handles.judulgrafik, 'String', 'Kurva
Pembangkitan');

function pushbutton2_Callback(hObject,
eventdata, handles)
set(handles.judulgrafik, 'String', 'Kurva
Optimasi');
grafikoptimasi=get(hObject, 'UserData');
plot(handles.axes2,
grafikoptimasi, 'LineWidth', 2);

function figure1_CloseRequestFcn(hObject,
eventdata, handles)
uiresume(hObject);

```


Halaman ini sengaja dikosongkan

RIWAYAT PENULIS



Penulis bernama lengkap M. Luthfiyansah. Anak terakhir dari tiga bersaudara. Kelahiran Lumajang tanggal 12 April 1993. Mengawali pendidikan di SD Islam Lumajang pada tahun 1999-2005 kemudian melanjutkan pendidikan pada tingkat menengah pertama di SMP Negeri 5 Lumajang pada tahun 2005-2008 setelah iu melanjutkan jenjang pendidikan di SMA Negeri 3 Lumajang pada tahun 2008-2011 selanjutnya penulis melanjutkan ke jenjang diploma di Institut Teknologi Sepuluh Nopember Surabaya, Fakultas Teknologi Industri, Jurusan D3 Teknik Elektro Industri pada tahun 2011-2015 dan kemudian melanjutkan ke jenjang sarjana di Institut Teknologi Sepuluh Nopember Surabaya, Fakultas Teknologi elektro Departemen Teknik Elektro program studi Teknik Sistem Tenaga 2015-hingga sekarang. Selama studi jenjang sarjana di ITS,

Email : luthfiyansah@gmail.com

Halaman ini sengaja dikosongkan